

**Defense Information Infrastructure (DII)  
Common Operating Environment (COE)**

**Programmer's Guide  
for  
Data Retrieval via METCAST**

**Document Version 1.5**

**09 November 2001**

**IMPORTANT NOTE:** FNMOC no longer offers technical support for customized use of the Metcast API's. This document will not be updated.

Prepared for:  
Space and Naval Warfare Systems Command  
METOC Systems Program Office (SPAWAR PMW-155)  
San Diego, CA

Prepared by:  
Fleet Numerical Meteorology and Oceanography Center  
Monterey, CA

and

Computer Sciences Corporation  
Monterey, CA



## **Table of Contents**

---

<b>1 SCOPE .....</b>	<b>1</b>
1.1 IDENTIFICATION.....	1
1.2 SYSTEM OVERVIEW .....	1
1.3 DOCUMENT OVERVIEW.....	2
<b>2 REFERENCED DOCUMENTS .....</b>	<b>3</b>
2.1 GOVERNMENT DOCUMENTS.....	3
2.2 NON-GOVERNMENT DOCUMENTS.....	4
<b>3 METCAST REQUEST OBJECTS .....</b>	<b>5</b>
3.1 THE MBL REQUEST LANGUAGE.....	5
3.1.1 Request Language Syntax.....	5
3.1.2 Request Syntax.....	6
3.1.2.1 General Structure of a Request .....	6
3.1.2.2 Parameters.....	7
3.1.3 Product Request Cookbook.....	14
3.1.3.1 Meteorological and Oceanographic Observations .....	14
3.1.3.2 Imagery Products .....	16
3.1.3.3 Gridded Products .....	16
3.1.3.4 Bulletins .....	17
3.1.3.5 Miscellaneous products.....	18
3.1.4 Examples.....	19
3.1.4.1 Basic examples.....	19
3.1.4.2 Several Bounding Boxes.....	20
3.1.5 Which Products are Available .....	21
3.2 THE MAILCAP OBJECT .....	21
3.3 THE CONFIGURATION OBJECT .....	23
3.3.1 Authorization and Access Authentication .....	25
<b>4 USING THE OMNICAST RETRIEVER TO RETRIEVE DATA.....</b>	<b>27</b>
4.1 USING THE OMNICAST RETRIEVER JAVA APIs.....	27

4.1.1	Prerequisites .....	27
4.1.2	Creating a Program That Connects With the Omnicast Retriever .....	29
4.1.3	Creating a Retrieval Session .....	31
4.1.4	Configuring Retriever Session Properties.....	31
4.1.4.1	Configuring the Retrieval Session to Use Authentication .....	32
4.1.4.2	Configuring the Retrieval Session to use a Proxy Server .....	33
4.1.4.3	Setting the Request Message for the Retrieval Session .....	34
4.1.4.4	Configuring the Mailcap Object for the Retrieval Session .....	34
4.1.5	Starting the Retrieval Session .....	35
4.2	RUNNING THE RETRIEVER USING THE COMMAND LINE INTERFACE	35
<b>5</b>	<b>FORMATS FOR RETRIEVED DATA .....</b>	<b>37</b>
5.1	DEALING WITH GRIDDED DATA .....	37
5.2	DEALING WITH OBSERVATION DATA .....	38
5.2.1	OMF Rationale.....	38
5.2.2	Structure of an OMF Document .....	39
5.2.2.1	Basic Attributes.....	40
5.2.2.2	The Reports Element .....	42
5.2.2.3	The Forecasts Element .....	43
5.2.2.4	The Advisories Element.....	43
5.2.3	OMF Markup for Each Report Type .....	44
5.2.3.1	OMF Markup for Observations in METAR Format.....	44
5.2.3.2	OMF Markup for SPECI Observations.....	45
5.2.3.3	OMF Markup for Synoptic (SYN) Observations.....	46
5.2.3.3.1	The SYID Element.....	48
5.2.3.3.2	The SYCODE Element.....	48
5.2.3.3.3	The SYG Element.....	49
5.2.3.3.4	The SYSEA Element .....	51
5.2.3.3.5	Synoptic Report Examples.....	53
5.2.3.4	OMF Markup for Pilot Weather Reports (PIREPS) .....	53
5.2.3.4.1	The PIID Element: .....	54
5.2.3.4.2	The PIC Element:.....	55
5.2.3.4.3	The PIRMKS Element .....	55

5.2.3.4.4	Examples of complete PIREPs .....	57
5.2.3.5	OMF Markup for Upper Air Reports.....	58
5.2.3.5.1	The UAPART Element.....	59
5.2.3.5.2	The UAID Element .....	60
5.2.3.5.3	The UACODE Element.....	60
5.2.3.5.4	The UALEVELS Element.....	61
5.2.3.5.5	The UALEVEL Element .....	64
5.2.3.5.6	Example of a Complete UAR Report .....	65
5.2.3.6	OMF Markup for Bathythermal, Salinity, and Current Reports (BTSC) .....	68
5.2.3.6.1	The BTID Element .....	70
5.2.3.6.2	The BTCODE Element.....	71
5.2.3.6.3	The BTLEVELS Element.....	71
5.2.3.6.4	The BTAIR Element.....	72
5.2.3.6.5	The BTLEVEL Element .....	73
5.2.3.6.6	Example of a complete BTSC Report.....	74
5.2.3.7	OMF Markup for a Terminal Aerodrome Forecast (TAF) .....	75
5.2.3.7.1	The VALID Element .....	76
5.2.3.7.2	The PERIOD Element.....	77
5.2.3.7.3	The PREVAILING Element .....	77
5.2.3.7.4	The VAR Element.....	78
5.2.3.7.5	Example of a Complete TAF Report .....	79
5.2.3.8	OMF Markup for SIGMETs .....	79
5.2.3.8.1	The VALID Element .....	80
5.2.3.8.2	The AFFECTING Element .....	80
5.2.3.8.3	The EXTENT Element.....	81
5.2.3.8.4	The BODY Element .....	81
5.2.3.8.5	Example of a Complete SIGMET Report.....	82
5.2.3.9	OMF Markup for an AIRMET Report.....	82
5.2.3.9.1	The HEADER Element.....	83
5.2.3.9.2	The VALID Element.....	84
5.2.3.9.3	The AIPART Element.....	84
5.2.3.9.4	Example of a Complete AIRMET Report .....	85

---

5.2.3.10	OMF Markup for Plain Text Messages .....	86
5.2.3.10.1	The MSG Element.....	87
5.2.3.10.2	Example of a Complete Messages Collection .....	89

**APPENDIX A        PARAMETER SPECIFICATIONS ..... A-1**

A.1	block_id.....	A-1
A.2	bounding-box .....	A-1
A.3	call_id.....	A-1
A.4	center-id and subcenter-id.....	A-2
A.5	depth-max .....	A-3
A.6	depth-min .....	A-3
A.7	product-id.....	A-4
A.8	layer.....	A-4
A.9	model.....	A-7
A.10	product-GRIB-code.....	A-9
A.11	tau.....	A-15
A.12	resolution.....	A-15

**APPENDIX B        THE COMPLETE OMF DTD ..... B-1**

**APPENDIX C        SAMPLE JAVA PROGRAM FOR DATA RETRIEVAL ..... C-1**

**APPENDIX D        CODE TO COMPUTE AN AUTHORIZATION CREDENTIAL ..... D-1**

---

**List of Tables**

---

**TABLE 3-1. MIME TYPES FOR COMMON DATA TYPES.....22**

**TABLE 5-1. BASIC ATTRIBUTES OF AN OBSERVATION IN OMF .....40**

<b>TABLE 5-2. OMF ATTRIBUTES FOR METAR AND SPECI REPORTS .....</b>	<b>44</b>
<b>TABLE 5-3. OMF ATTRIBUTES FOR THE SYN ELEMENT.....</b>	<b>46</b>
<b>TABLE 5-4. OMF ATTRIBUTES FOR THE SYG ELEMENT .....</b>	<b>49</b>
<b>TABLE 5-5. OMF ATTRIBUTES FOR THE SYSEA ELEMENT.....</b>	<b>52</b>
<b>TABLE 5-6. OMF ATTRIBUTES FOR THE PIREP REPORTS.....</b>	<b>53</b>
<b>TABLE 5-7. OMF ATTRIBUTES FOR THE PIC ELEMENT.....</b>	<b>55</b>
<b>TABLE 5-8. OMF ATTRIBUTES FOR THE UALEVEL ELEMENT .....</b>	<b>64</b>
<b>TABLE 5-9. OMF ATTRIBUTES FOR THE BTSC ELEMENT .....</b>	<b>69</b>
<b>TABLE 5-10. OMF ATTRIBUTES FOR THE BTID ELEMENT .....</b>	<b>70</b>
<b>TABLE 5-11. OMF ATTRIBUTES FOR THE BTAIR ELEMENT .....</b>	<b>73</b>
<b>TABLE 5-12. OMF ATTRIBUTES FOR THE BTLEVEL ELEMENT .....</b>	<b>73</b>
<b>TABLE 5-13. OMF ATTRIBUTES FOR THE TAF ELEMENT.....</b>	<b>75</b>
<b>TABLE 5-14. OMF ATTRIBUTES FOR THE SIGMET ELEMENT.....</b>	<b>79</b>
<b>TABLE 5-15. OMF ATTRIBUTES FOR THE EXTENT ELEMENT .....</b>	<b>81</b>
<b>TABLE 5-16. OMF ATTRIBUTES FOR THE AIRMET ELEMENT .....</b>	<b>82</b>
<b>TABLE 5-17. OMF ATTRIBUTES FOR THE MSG ELEMENT .....</b>	<b>87</b>
<b>TABLE A-1. CENTER-ID AND SUBCENTER-ID CODES.....</b>	<b>A-2</b>
<b>TABLE A-2. PRODUCT-ID CODES .....</b>	<b>A-4</b>
<b>TABLE A-3. LAYER PARAMETER CODES .....</b>	<b>A-4</b>
<b>TABLE A-4. MODEL PARAMETER CODES .....</b>	<b>A-7</b>

**TABLE A-5. PRODUCT-GRIB-CODE AND PRODUCT KEYWORD VALUESA-9**

---

**List of Figures**

---

Figure 2-1. METCAST Data Flow .....2

# 1 SCOPE

## 1.1 IDENTIFICATION

This document provides guidance to programmers who wish to use METCAST to retrieve data for use in their programs. It includes an introduction to METCAST, information concerning the METCAST request language and how to use it to formulate a request, suggestions for setting up *mailcap* and *config* files for the retrieval, a discussion of ways to initiate a retrieval session, and information about the file formats delivered by METCAST.

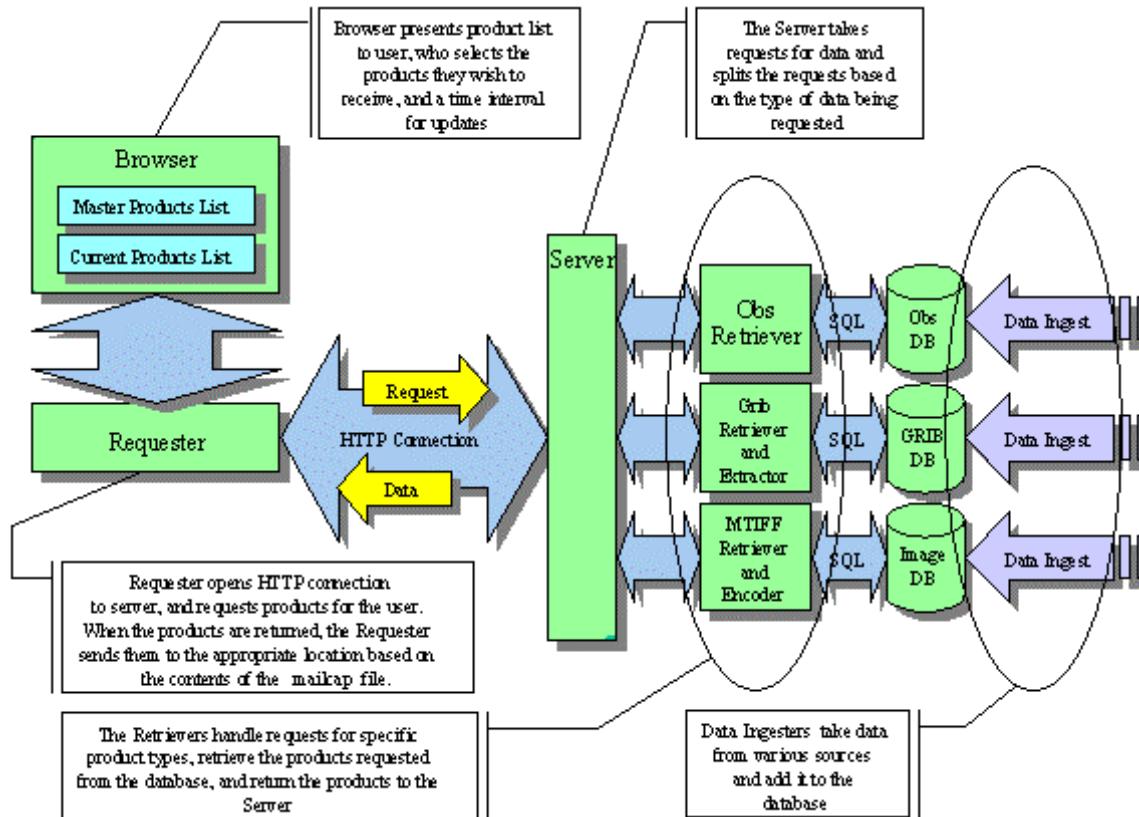
While METCAST currently provides no Application Program Interfaces (APIs) for requesting and receiving data directly, it does provide methods to initiate a retrieval with specified request and configuration files, and for communicating with the retriever to monitor retrieval status. Sections 3 through 5 of this document detail three main steps in the retrieval process:

1. Specifying the data to be retrieved and retriever configuration (Section 3),
2. Initiating and monitoring a retrieval (Section 4), and
3. Dealing with the downloaded data (Section 5).

## 1.2 SYSTEM OVERVIEW

**METCAST** is a standards-based, request-reply and subscription system used to transmit data across the Web using HTTP (Hyper-Text Transfer Protocol). METCAST uses a client-server architecture in which a server, connected to a METOC database, publishes a dynamic product list showing all data currently in the system and all channels available through the server. Clients subscribe to the product list automatically when their connection to a server is active, and thus continually maintain a list of data available on the system. A client may be connected to multiple servers, and will maintain a separate product list for each server.

When a request is scheduled, the METCAST Client formulates a request message and forwards it via HTTP to the server(s) from which the data are requested. The server checks its database to find out whether it has any new data to fill the request. If not, it returns a message to that effect. If there is new data, the server extracts the data from its database, packages it, and returns it to the client. Figure 2 shows this data flow.



**Figure 2-1. METCAST Data Flow**

## 1.3 DOCUMENT OVERVIEW

The remainder of this document is organized as follows:

- Section 2 Provides a list of applicable documentation
- Section 3 Discusses METCAST request objects. Issues discussed include the METCAST request language, *mailcap* objects (which determine the handling of downloaded data), and configuration objects (which determine the frequency and timing of data downloads).
- Section 4 Discusses the facilities provided in the Omnicast Retriever for creating request objects and for initiating and monitoring a METCAST retrieval session from various types of programs.
- Section 5 Deals with the file and data formats in which METCAST downloads are packaged. Different types of data are supplied in different formats.
- Section 6 Provides applicable notes, including a glossary of acronyms.

## **2 REFERENCED DOCUMENTS**

### **2.1 GOVERNMENT DOCUMENTS**

Commander, Naval Meteorology and Oceanography Command

COMNAVMETOCOMINST 3141.2      *Surface METAR Observations Users Manual*

COMNAVMETOCOMINST 3144.1D      *Ship Weather Observations Manual*

Office of the Federal Coordinator for Meteorological Services and Supporting Research,  
Washington, DC

FMH-2                  *Federal Meteorological Handbook No. 2: Surface Synoptic Codes*  
December 1988

FMH-10                *Federal Meteorological Handbook No. 10: Meteorological Rocket  
Observations*  
December 1988

FMH-11                *Federal Meteorological Handbook No. 11: Doppler Radar Meteorological  
Observations*  
June 1991

FMH-1                 *Federal Meteorological Handbook No. 1: Surface Weather Observations  
and Reports*  
December 1995

Fleet Numerical Meteorology and Oceanography Center, Monterey, CA

Unnumbered            *A Guide to MBL -- A Metcast Request Language.* Available online at  
5 October 1999        <http://zowie.metnet.navy.mil/~spawar/JMV-TNG/Request-Lang.html>

Unnumbered            *Weather Observation Definition Format.* Available online at  
14 October 1999       <http://zowie.metnet.navy.mil/~spawar/JMV-TNG/XML/OMF.html>

Unnumbered            *Metcast MIME Type List.* Available online at  
<http://zowie.metnet.navy.mil/~spawar/mailcap.txt>

Unnumbered            *Distributing Weather Products Through an HTTP Pipe.* Available online at  
5 October 1999        <http://zowie.metnet.navy.mil/~spawar/Briefs/Jmv-tng/Metcast-Intro.html>

## **2.2 NON-GOVERNMENT DOCUMENTS**

### World Meteorological Organization (WMO), Geneva, Switzerland

WMO-WWW Tech Report 17 May 1994      *Guide to WMO Binary Code Forms* (Part1 deals with BUFR, Part 2 with GRIB). Available online at <http://www.wmo.ch/web/www/reports/Guide-binary.html>

WMO 306  
1995                  *Manual on Codes*

### World Wide Web Consortium (W3C)

RFC 2068  
January 1997                  *Hypertext Transfer Protocol -- HTTP/1.1*

## 3 METCAST REQUEST OBJECTS

A program needs to create three objects before initiating a request for data via METCAST:

1. A request object specifying the data to be retrieved, formatted in the METCAST Request Language (MBL) described in Section 3.1,
2. A *mailcap* object specifying the handling of the downloaded data, as specified in Section 3.2, and
3. A configuration object specifying the frequency and timing of the download (this is optional, and need not be specified for one-time, on-demand retrievals). Section 3.3 discusses the configuration object.

Depending on the interface chosen, these objects may be stored in files or created programmatically as required. The Omnicast Retriever provides interfaces for creating these objects programmatically, as will be shown in Section 4 of this document.

### 3.1 THE MBL REQUEST LANGUAGE

This section describes the language a Metcast client uses to communicate its request to a server. MBL phrases constitute the body of a Metcast client request object.

#### 3.1.1 Request Language Syntax

The syntax of the language for requesting weather products has a very simple structure to parse and to interpret.

The request language is built around a single syntactic category: an *S-expression* similar to expressions used in LISP and similar languages. The syntax is free-form, clear, and easily understood both by a human being and the computer. An S-expression is delimited by a pair of parentheses.

An S-expression may be:

- An empty expression, denoted as ( ).
- A pair of parentheses containing an *atom* (see below) or another S-expression. If x is an atom or another S-expression, then (x) is an S-expression.
- If s<sub>1</sub> and s<sub>2</sub> are two S-expressions, so is their union (merge), which is made by concatenating the two S-expressions, erasing touching parentheses and possibly introducing spaces to keep their tokens from coalescing.

Components of the S-expression are kept separate with white spaces (blanks, HT and newlines). The number of white spaces is irrelevant, and can be used to improve readability.

An *atom* is

- A number (integer or fixed/floating point), in the obvious format: anything that looks like a number to a C compiler, is a number to us. Only radix-10 numbers are currently supported.
- A lat-spec, a signed integer or a fixed-point number, optionally followed by a letter 'N' or 'S' (for Northern and Southern hemispheres). The number is interpreted as degrees latitude, with positive values referring to the Northern hemisphere. The number must be within [-90, 90]. Examples: 45N, 45.0, -50.1, 50.1S
- A lon-spec, a signed integer or a fixed-point number, optionally followed by a letter 'E' or 'W' (for Eastern and Western hemispheres). The number is interpreted as degrees longitude, with positive values referring to the Eastern hemisphere. Examples: 145E, 145.0, -10.1, 10.1W, 180.0
- An HTTP token, as defined in [RFC2068](#). Basically, an HTTP token is a sequence of token characters A-Za-z0-9!#\$%&!\*+-.^\_`|~
- A double-quoted string, which may include any printable character; if a double quote occurs inside the string, it must be quoted by preceding it with a backslash character; any backslash character that is a part of the string should be doubled.

An MBL request consists of an outer S-expression containing the entire request, with individual S-expressions denoting the area(s) of interest, product types, product specifications, etc nested within.

### 3.1.2 Request Syntax

#### 3.1.2.1 General Structure of a Request

An MBL request has the following general structure:

```
(area-id parameter ...
  (products product-desc ...) )
```

where *area-id* is a geographic area identifier used as a transaction id. It can be anything a client chooses to identify its request, for example, to match several requests to the corresponding replies. *area-ids* are specific to a client, so different clients may choose the same ids for their transactions without clashing.

Each *product-desc* has the following format

(*product-id parameter ...*)

where *product-id* identifies a product being requested. The *parameters* are arguments that specialize the request. The cookbook section below lists ids of all the typical products as well as their mandatory and optional arguments.

When a Metcast server is about to fulfill a request for a product, it looks for the product arguments. The server first searches the list of the parameters specified in that product's scope. If the search fails the server scans parameters in the global request scope. It makes sense therefore to place the most common arguments (eg, *bounding-box*) into the global scope, and specify particular or overriding parameters by the corresponding product. See the cookbook for more details.

### 3.1.2.2 Parameters

parameters are one or more of the following. They may be specified in an arbitrary order. Each parameter is contained within an S-expression.

#### **block\_id**

(*block-id block-id-num*)

Specifies the WMO-assigned Block Number and Station Number for a reporting station. As interpreted by the MBL interpreter, this is a 1 to 6 digit numerical identifier.

Example: (block\_id 72491)

#### **bounding-box**

(*bounding-box N-LAT W-LON S-LAT E-LON*)

Specifies the latitudinal and the longitudinal spans of the area. Here *N-LAT* is the latitude of the northernmost point of the area, *S-LAT* is the latitude of the southernmost point, *W-LON* is the longitude of the westernmost point of the area, and *E-LON* is the easternmost longitude.

Examples:

(bounding-box 70.0N 25.00W -50 175)

(bounding-box 90 -180 -90 180)

It is possible to define multiple bounding boxes in a single request. In the example below, a bounding box is defined for each product requested:

```
(CAU
(bounding-box 47.0 -12.0 29.0 40.0)
(products
(grib (product-GRIB-code 7) (layer isobar 1000) (tau 0)
(source 58) (process-id 22) (resolution 0.200))
(grib
(bounding-box 47.5 -12.5 27.5 40.0)
(product-GRIB-code 7) (layer isobar 900) (tau 0) (source 7)
(process-id 80) (resolution 2.500))))
```

Here a default bounding box is established in the second line (in italics). This bounding box will apply to all products that do not contain a bounding box entry of their own (i.e., this is the default for all products that don't define their own bounding boxes). The definition of products then begins; the first (`grib` entry defines the first product to be retrieved. The second (`grib` entry defines a new bounding box (which applies to this product only) and another product.

### **call\_id**

(`call-id call-letters`)

This parameter provides the International Civil Aviation Organization (ICAO) call sign for a land station, an alphanumeric ship call sign for a reporting ship, or a buoy serial number. When retrieving bulletins, this parameter represents the ICAO call sign of the issuing station. The call sign is a character string.

Examples:

(`call_id "KMRY"`)  
(`call_id " WST9756"`)

### **center-id**

(`center-id center-id`)

Identifies the production center that originated a grid product, (e.g. FNMOC, US Weather Service - National Met. Center) (see Appendix A, [Section A.4](#), WMO-WWW Tech Report 17, Part 2, Table 0, or WMO 306, Volume 1.2, Part C, Table C-1). Defaults to 58, which is the identifier for FNMOC.

### **depth-max**

(`depth-max depth-max-limit`)

Depth, in meters, for the bottommost vertical layer to be returned by a query (e.g., a BTSC product query).

Example: (`depth-max 1500`)

### **depth-min**

(`depth-min depth-min-limit`)

Depth, in meters, for the topmost vertical layer to be returned by a query (e.g., a BTSC product query).

Example: (`depth-min 10`)

### **grid-id**

Identifies the grid for gridded products. See Appendix A, [Section A.9](#) for a list of grid-ids currently in use.

**isobar-p-max**

```
(isobar-p-max isobar-p-max-limit)
```

This is the pressure, in hPa, for the bottommost vertical layer to be returned by a query (e.g., a UAR product query). The default is the surface pressure at the station.

Example: `(isobar-p-max 975)`

**isobar-p-min**

```
(isobar-p-min isobar-p-min-limit)
```

This is the pressure, in hPa, for the topmost vertical layer to be returned by a query (e.g., a UAR product query). The default is the highest layer present in the station's sounding.

Example: `(isobar-p-min 10)`

**layer**

```
(layer layer-spec)
```

This phrase is used to ask for a specific level of a product: at a particular height above the ground, at a particular isobar level, at `msl` (mean sea level), between two isobar layers, etc. See Appendix A, [Section A.8](#) or Tables 3 and 3a of WMO-WWW Tech Report 17, Part 2 for more details.

**manops**

```
(manops manop1 manop2 ...)
```

Specifies Manual of Operations (MANOP) headers for bulletins to be retrieved. The MANOP header format is specified in WMO Publication 386 as  $T_1T_2A_1A_2ii$  CCCC YYGGgg, where  $T_1$  and  $T_2$  represent the type and subtype of information contained in the bulletin,  $A_1A_2$  represents the geographic area of coverage, and  $ii$  is a sequential number for the issuing station. The CCCC group gives the ICAO call sign of the issuing station, and YYGGgg represents the day of the month (YY) and the hour and minute of issuance (GGgg) of the bulletin.

The `manops` parameter must be specified as a list of strings, each of which may be:

- A complete  $T_1T_2A_1A_2ii$  section of the MANOP header of the desired message; the first 4 characters must be uppercase letters and the last two digits must be numbers.  
Example: "FABA31".
- A partial message ID; an ID as above but containing SQL-style wildcards.  
Examples: "AX%", "ABAK%".
- The complete  $T_1T_2A_1A_2ii$  plus the call letters of the issuing station appended (without a separating space); e.g. "FAUS1KDFW".

**max-records**

```
(max-records output-limit)
```

This parameter specifies the maximum number of records to be output by a METCAST query.

Example: `(max-records 50)`

**mime-type**

```
(mime-type mime-type-spec)
```

Specifies the MIME type to tag the product content with. Each product served by a Metcast server has the default MIME type associated with it. The client, however, may override this default by setting this parameter.

**model**

```
(model model-id)
```

Where the `model-id` identifies the particular computation process that was used to produce gridded data. For example, NOGAPS, NOGAPS-1deg, NORAPS-CONUS, OTIS, etc. See the list of served models in Appendix A, [Section A.9](#).

Optional. The default value is NOGAPS-1deg.

**modified-since**

```
(modified-since epoch-timestamp)
```

This parameter specifies the earliest desired creation date for products requested. The value is an epoch time (time in seconds since 0000Z 1 January 1970). This parameter is used to exclude retrieval of products made earlier than the time specified (i.e. to limit the retrieval to products made since the specified time). If the value is zero (the default), products will be retrieved regardless of their age.

A METCAST server implicitly inserts the `modified-since` parameter into the global request scope based on a `IF_MODIFIED_SINCE` HTTP header. Therefore a client does not need to mention this parameter in his request. It may make sense however to place a `modified-since` argument into the scope of a particular product request. For example, a user may say

```
(area ... (products (SIGMET (modified-since 0))))  
to request all rather than the latest SIGMET warnings, even in a conditional request.
```

**msgtype**

```
(msgtype "T1T2")
```

Specifies the bulletin type and subtype when retrieving bulletins. This is a string corresponding to the T<sub>1</sub>T<sub>2</sub> portion of the bulletin's MANOP header.

Example: (msgtype "FE") would retrieve all extended forecast bulletins.

**process-id**

```
(process-id num-process-id)
```

Specifies the numerical model which produced gridded data. See Appendix A, [Section A.9](#) for a list of process-ids currently in use. The default is 58 (NOGAPS).

Example: (process-id 43)

**product-GRIB-code**

```
(product-GRIB-code grib-code-value)
```

Specifies the code for a grib product. See Appendix A, [Section A.10](#) for a list of codes supported as of this writing, or Table 2 of WMO-WWW Tech Report 17, Part 2, or a dynamic list of available products. For example, (product-GRIB-code 1) requests a pressure grid.

This parameter is mandatory if a product with a grib product-id is requested.

Example: (product-GRIB-code 11)

**productname**

```
(productname productname-value)
```

Specifies the name of a satellite product to retrieve. This is a mandatory parameter for imagery products.

Example: (productname "N.Americ\_visible\_FNMOC.jif")

**products**

```
(products product-desc ...)
```

Each *product-desc* has the following format

```
(product-id parameters ...)
```

where *product-id* identifies a particular product. See Appendix A, [Section A.1](#) for a list of acceptable product-ids. The *parameters*, which are generally only needed for gridded and satellite products, tell what else is needed to specify a product. See Appendix A for a list of product-ids currently available, and lists of parameters currently in use.

**resolution**

```
(resolution x-res y-res) or  
(resolution grid-spacing)
```

Where the `grid-spacing` is the a floating point number specifying the grid mesh size in degrees. Currently only the second format is supported.

Optional. The default value is 1.0 degree, but may be overridden by a particular model.

Example: `(resolution 0.5)`

**source**

```
(source center-id)
```

Where `center-id` identifies the production center originating a grid product. This ID can either be a symbol -- eg, FNMOC, AFGWC, NWSTG -- or a number. In the latter case, the number is assumed to be `100*num_subcenter_id + numerical_center_id`. For example, `(source 7)` and `(source NCEP)` both refer to US Weather Service - National Met. Center, while `(source 107)` means NCEP-Re-analysis project; `(source 57)` refers to the USAF Global weather center, and `(source 1057)` means USAF TBD. If the subcenter id is zero the `center-id` number corresponds to the center id as defined in WMO-WWW Tech Report 17, Part 2, Table 0, part 1.

The `source` specification is optional. The default value is 58, which is the ID for FNMOC.

The `center-id` and `subcenter-id` parameters may be substituted for the `source` parameter. Each of these parameters takes an integer number (e.g. `(center-id 58)` (`subcenter-id 0`) for FNMOC).

**st\_constraint**

```
(st_constraint a-st-constraint1 a-st-constraint2...)
```

Limits choices to particular station(s) or countries. Each `a-st-constraint` may be a WMO block station ID (`block_id`), ICAO call sign (`call_id`), station name (`st_name`), or country code (`st_country_code`).

Example: `(st_constraint (call_id "KMRY"))`

**st\_country\_code**

```
(st_country_code st-country-code)
```

WMO country code for the station: a string of two uppercase letters.

Example: `(st_country_code "US")`

**st\_country\_name**

(*st\_country\_name st-country-name*)

Full uppercase name for the station's country. It may contain SQL-style wildcards '\_' and '%'.

Example: (*st\_country\_name "UNIT%"*)

**st\_name**

(*st\_name st-name*)

The uppercase name for the station. It may contain SQL-style wildcards '\_' and '%'.

Example: (*st\_name "MONTEREY"*)

**sub-center id**

(*sub-center-id subcenter-id*)

Center-specific identifier for the subcenter that originated a grid product (see Appendix A, [Section A.4](#))

**tau**

(*tau tau-value1 tau-value2 ...*)

Asks for forecast data at *tau-value* hours into the future. This parameter takes one or more integer values, separated by spaces, e.g. (*tau 0 12 24 48 72*) requests the analysis (*tau 0*) and the 12, 24, 48, and 72 hour forecasts.

**time-mod-since**

(*time-mod-since time-interval*)

Specifies the time interval in minutes during which a message must have been modified in order to be retrieved. This allows the user to only request messages that have been updated within a certain time period.

Example: (*time-mod-since 120*) only retrieves message modified within the last 2 hours (120 minutes).

**valid-at**

(*valid-at epoch-timestamp*)

This parameter, if present, limits the query to only those products (e.g., forecasts or warnings) which are still valid as of *epoch-timestamp*, which is an epoch time (seconds since 0000Z 1 January 1970). This time moment may point into the future as well as into the past.

### **3.1.3 Product Request Cookbook**

This section shows the most common product types and the parameters that may be specified for each one. Note that certain parameters are required; that is, they must always be specified in requests for the associated product type. Required parameters are specified in bold type.

#### **3.1.3.1 Meteorological and Oceanographic Observations**

The products are returned as OMF XML formatted documents, described in Weather Observation Definition Format (OMF)

<b>Product Id</b>	<b>Parameters (mandatory are in bold)</b>
<b>MSL</b>	<b>bounding-box</b> modified-since mime-type max-records block_id call_id st_name st_country_code st_country_name
Query of the Master Station Library to get information about one or more reporting stations	
<b>METAR</b>	<b>bounding-box</b> modified-since mime-type st_constraint
Hourly, special, and synoptic surface weather observation reports from land stations (METAR, SPECI, SYNOP)	
<b>PIREPS</b>	<b>bounding-box</b> modified-since mime-type st_constraint
PILOT Reports. Inflight weather observations filed by pilots at unscheduled times.	

<b>SYNSEA</b>	<b>bounding-box</b> modified-since mime-type call_id
Surface synoptic observations from sea stations (moving or drifting or moored)	

<b>UAR</b>	<b>bounding-box</b> modified-since mime-type st_constraint isobar-p-min isobar-p-max
Rawinsonde and Pilot Balloon (PIBAL) Observations (Upper Air Reports)	

<b>BTSC</b>	<b>bounding-box</b> modified-since mime-type call_id depth-min depth-max
Bathythermal, Salinity and Current Observations (BATHY, TESAC, TRACKOB)	

<b>TAF</b>	<b>bounding-box</b> modified-since mime-type st_constraint valid-at
Terminal Aerodrome Forecasts	

<b>SIGMET</b>	<b>bounding-box</b> modified-since mime-type valid-at
SIGMET advisories	

<b>AIRMET</b>	<b>bounding-box</b> modified-since mime-type valid-at
AIRMET advisories (AIRman's METeorological Information)	

### 3.1.3.2 Imagery Products

The products are usually returned in TIFF (or Metoc TIFF: MTIFF, MIF or JIF) formats.

<b>Product Id</b>	<b>Parameters (mandatory are in bold)</b>
<b>imagery</b>	<b>bounding-box</b> modified-since <b>productname</b>
Request for a satellite image	

### 3.1.3.3 Gridded Products

Gridded data are sent in a GRIB format.

<b>Product Id</b>	<b>Parameters (mandatory are in bold)</b>
<b>grib</b>	<b>bounding-box</b> modified-since <b>product-GRIB-code</b> source model center-id subcenter-id process-id grid-id resolution layer tau
Request for a grid product	

<b>pressure</b>	<b>bounding-box</b> modified-since source model center-id subcenter-id process-id grid-id resolution layer tau
Request for a pressure grid. There is a one-to-one correspondence between GRID product names and product-GRIB-codes; see Appendix A, Table A-4. For example, the following two requests are equivalent:	

```
(products (pressure (model "NOGAPS") (layer isobar 1000)))  
  
(products (grib (product-GRIB-code 1)  
                (model "NOGAPS") (layer isobar 1000)))
```

### 3.1.3.4 Bulletins

Product Id	Parameters <b>(mandatory are in bold)</b>
<b>Messages</b>	<b>manops</b> call_id msgtype keywords lat-n lat-s lon-e lon-w time-mod-since
Free-text bulletins.	

### 3.1.3.5 Miscellaneous products

Product Id	Template
<b>GlobalWarnings</b>	(GlobalWarnings <i>warning-id1</i> <i>warning-id2</i> ...)

Free-text warnings and advisories pertaining to the entire globe (or without any specific geographical registration).  
*warning-id* can be:

trop-cyc  
An advanced tropical cyclone warning

jots-w  
A JOTS (high wind, etc) warning

<b>DiscreteThings</b>	(DiscreteThings <i>channel-id1</i> <i>channel-id2</i> ...)
Requesting a <i>thing</i> -- an unstructured document -- from a Metcast Channel. See Metcast Channels for more details.	

### 3.1.4 Examples

#### 3.1.4.1 Basic examples

##### A single grid request

```
(NORF
  (bounding-box 45.0N 99W 23.0 74W)
  (model NOGAPS)
  (products
    (pressure (layer msl) (tau 24))))
```

##### Asking for several gridded products

```
(some-area
  (bounding-box 90.0 178W -90.0 179E)
  (source FNMOC) (model NOGAPS)(resolution 1.0)
  (products
    (geopotential-height (layer isobar 500)(tau 12))
    (temperature (layer msl))))
(some-area
  (bounding-box 90.0 178W -90.0 179E)
  (source FNMOC) (model NOGAPS)(resolution 1.0)
  (products
    (geopotential-height (layer isobar 500)(tau 12))
    (grib (product-GRIB-code 55) (layer msl) (tau 12) (source 7))
    (grib (product-GRIB-code 155) (layer msl) (tau 24) (source
15557))))
```

##### Satellite imagery request

```
(BIGarea
  (bounding-box 70.1N 25.00W -50 175)
  (products
    (imagery (productname "bayvis.mif"))))
```

##### A request for observation data (METAR), warnings, and SIGMETS

```
(Some-area
  (bounding-box 30.0N 5.00W -5 20)
  (products
    (GlobalWarnings trop-cyc jots-w)
    (METAR)))
(ReqID
  (bounding-box 56.6N 122.0W 10.5 70.5W)
  (products (SIGMET (valid-at 916993601))))
```

##### Asking for METARS and TAFS from a specific station

```
(ReqID (bounding-box 37N 122.0W 36.5 121.5W)
  (products (METAR (st_constraint (call_id "KMRY")))))
  (products (Forecasts (st_constraint (call_id "KMRY"))))))
```

## Requesting specific bulletins

```
(ReqID (bounding-box 89N 125W 1N 55W)
      (products (Messages (manops WTNT%))))
```

## Requesting a few *things*

```
(ReqID
  (products
    (DiscreteThings 10 14)))
(Area51
  (use special-area-51))
(Area51
  (use special-area-51) (user 007)
  (products (TAF)))
```

as long as you and the server agree on the defaults, `special-area-51` and `007` in the case above.

### 3.1.4.2 Several Bounding Boxes

An example below requests essentially the same product but from two different weather models: COAMPS and NCEP. The `bounding-box` for the latter model has to be adjusted to reflect that model's coarser resolution.

```
(CAU
  (bounding-box 47.0 -12.0 29.0 40.0)
  (products
    (grib (product-GRIB-code 7) (layer isobar 1000) (tau 0)
          (source 58) (process-id 22) (resolution 0.200))
    (grib
      (bounding-box 47.5 -12.5 27.5 40.0)
      (product-GRIB-code 7) (layer isobar 900) (tau 0) (source NCEP)
      (process-id 80) (resolution 2.500))))
```

Note the two `bounding-boxes`. The first one is the *default* bounding box, which applies to all the products that do not specify their own. The second `bounding-box` is for the NCEP product only. There is only one `products` clause (as it should be).

In the following example the area of interest is made of two distinct, non-overlapping geographical regions:

```
(CAU
  (bounding-box 47.0 -12.0 29.0 40.0)
  (products
    (grib (product-GRIB-code 7) (layer isobar 1000) (tau 0)
          (source 58) (process-id 22) (resolution 0.200))
    (METAR
      (bounding-box 38 -123 36 -121))))
```

Executing this request retrieves a GRIB file for an area in Europe, and METARs for the San Francisco Bay area. This particular request might not be too useful, yet it makes a fine example.

### **3.1.5 Which Products are Available**

Appendix A provides a list of `product-ids` and other parameters in use at the time this document was published. This list is dynamic and subject to change.

Any Metcast client may query a server for a list of products it serves (and which of them are potentially or actually available). The server returns a list, in XML format, that shows all areas and products known to the server.

The `Abstract` field in this list is intended to be user-friendly, to be used by a client to describe a product to a user. The `Title` and `Usage` fields tell the client how to request that particular product.

## **3.2 THE MAILCAP OBJECT**

The mailcap object is used to specify the processing of the data downloaded during a retriever session. The options are specified according to the Multipurpose Internet Mail Extensions (MIME) content type, which will be provided in each set of data returned by the retriever.

The syntax of a mailcap object is quite simple. Any line that starts with "#" is a comment. Blank lines are ignored. Otherwise, each line defines a single mailcap entry for a single content type. Long lines may be continued by ending them with a backslash character, \.

Each individual mailcap entry consists of a content-type specification, a command to execute, and (possibly) a set of optional "flag" values. For example, a very simple mailcap entry would look like this:

```
text/plain; cat %s
```

The optional flags can be used to specify additional information about the command. For example:

```
text/plain; cat %s; copiousoutput
```

can be used to indicate that the output of the 'cat' command may be voluminous, requiring either a scrolling window, a pager, or some other appropriate coping mechanism.

The `type` field (`text/plain`, in the above example) is simply any legal content type name, as defined by RFC 822. In practice, this is almost any string. It is the string that will be matched against the `Content-type` header (or the value passed in with -c) to decide if this is the mailcap entry that matches the current message. Additionally, the `type` field may specify a subtype (e.g. `text/ISO-8859-1`) or a wildcard to match all subtypes (e.g. `image/*`).

The `command` field is any UNIX command ("`cat %s`" in the above example), and is used to specify the interpreter for the given type of message. It will be passed to the shell via

the system facility. Semicolons and backslashes within the command must be quoted with backslashes. If the command contains %s, those two characters will be replaced by the name of a file that contains the body of the message. If it contains %t, those two characters will be replaced by the *content-type* field, including the subtype, if any. (That is, if the content-type was `image/pbm; opt1=something-else`, then %t would be replaced by `image/pbm`.) If the command field contains %{} followed by a parameter name and a closing {}, then all those characters will be replaced by the value of the named parameter, if any, from the *Content-type* header. Thus, in the previous example, %{}opt1} will be replaced by something-else. Finally, if the command contains \%s, those two characters will be replaced by a single % character. (In fact, the backslash can be used to quote any character, including itself.

If no %s appears in the command field, then instead of placing the message body in a temporary file, the body will be passed to the command on the standard input. This is helpful in saving /tmp file space, but can be problematic for window-oriented applications under some window systems such as MGR.

Mutiple processes may be placed in the command field by separating them with pipe ('|) symbols, e.g.

```
image/x-mif; copy %s %{area}\%\%s | proCDATA %{area}\gridded.con %s  
%{productname} %{timestamp} PROCSAT
```

The METCAST implementation of *mailcap* also supports special <start></start> and <end></end> tag sets, whose use is optional. These tags specify preprocessing and postprocessing commands. Commands between the <start> and </start> tags are forced to the head of the line, so that they are always done before the commands in the main body. Commands between the <end> and </end> tags are performed after the commands in the main body. In the entry below, the main body command copies the data received to a file in the directory for the area's data. The command in the <end> tags runs the *proCDATA* program using the area's *gridded.con* file to specify configuration parameters and with the PROC3D option set to process a 3D data volume.

```
text/x-packed-eof; copy %s %{area}\%\%s <end>proCDATA  
%{area}\gridded.con PROC3D</end>
```

Table 3-1 below specifies the MIME content types returned for common data types.

**Table 3-1. MIME Types for Common Data Types**

Data Type	MIME Type
2D Gridded (model output) data	application/x-grib
3D Gridded data volume (EOF encoded)	application/x-packed-eof
GIF imagery	image/gif
TIFF imagery	image/tiff
METOC TIFF (MIF) imagery	image/x-mif

**Table 3-1. MIME Types for Common Data Types**

<b>Data Type</b>	<b>MIME Type</b>
HTML	text/html
Plain text	text/plain
Observation data (METARs and SPECIs)	text/x-omf
Observation data (SYNOP code)	text/x-omf-synop
Upper air observation data	text/x-omf-uar
Terminal Aerodrome Forecast (TAF) data	text/x-omf-forecasts
Significant weather advisories (SIGMETs)	text/x-omf-advisories
Warnings in OTHT-Gold format	text/x-oth-warnings
Tropical cyclone warnings	text/x-tc-warnings
Airmans Meteorological Information (AIRMETs)	text/x-omf-airmet
Pilot Reports (PIREPs)	text/x-omf-pirep

### **3.3 THE CONFIGURATION OBJECT**

This is the object (or file) that stores all configuration information the Omnicast Retriever needs to establish a connection to a HTTP server and submit a (properly authenticated) request for products and updates. The file is normally named <sessionname>.con (although any other name would do): this file is passed to the Retriever as the first command-line parameter.

The file adheres to the standard JMV config file format

#### **SERVER\_URL**

The *complete URL* to a Metcast server (a CGI script on a HTTP server that can take a Metcast request and try to handle it). This is the same kind of information one types on the "Location:" line in Netscape.

Examples:

```
SERVER_URL = http://152.80.34.202/cgi-bin/server
SERVER_URL = HTTP://nites-2:80/
SERVER_URL = http://www.nlmoc.navy.mil
SERVER_URL = file:///users/oleg/HTTP-Retrieve/conus.mime
```

#### **PROXY\_NAME**

The name (or an IP address) of a HTTP proxy. This is the same thing a user enters into Netscape's "Options:Network Preferences:Proxies:Manual Proxy Configuration:HTTP Proxy" field.

Examples:

```
PROXY_NAME = 192.16.167.45
```

PROXY\_NAME = phobos.metnet.navy.mil

#### **PROXY\_PORT**

The port on the HTTP proxy host for listening (and relaying) HTTP connections. This is the same value a user enters into Netscape's "Options:Network Preferences:Proxies:Manual Proxy Configuration:HTTP Proxy:Port" field. If this parameter is omitted, a default value of 80 is assumed.

Example:

PROXY\_PORT = 80

#### **PROXY\_ENABLE**

TRUE/FALSE: tells if the SERVER\_URL is to be accessed through the proxy specified by PROXY\_NAME and PROXY\_PORT. If PROXY\_ENABLE=FALSE the corresponding fields in the JMV net options panel ought to be grayed out.

#### **AUTH\_CREDENTIAL**

A credential of a particular user or site to submit to a HTTP server for authentication. The retriever as a user agent sends this credential to the server in a Authorization: request header. The credential may be required by the server to access a (supposedly restricted) resource specified in SERVER\_URL

Example:

AUTH\_CREDENTIAL = Basic Y2xhbXM6Y2hvd2Rlcg==

See [Section 3.3.1, Authorization and Access Authentication](#) for more details.

#### **AUTH\_ENABLE**

TRUE/FALSE: tells if the client wants to go into trouble of authenticating itself. If the resource at SERVER\_URL is publicly available (or a Metcast's Web server would grant access considering client's IP address), the authentication may not be necessary.

Again, if AUTH\_ENABLE = FALSE, USERNAME and PASSWORD fields on the JMV configuration panel ought to be grayed out; as far as the retriever is concerned, it won't even look for the AUTH\_CREDENTIAL field in this configuration file if AUTH\_ENABLE is FALSE or omitted.

#### **REQ\_IF\_MODIFIED**

TRUE/FALSE; FALSE means the client is asking a Metcast server for products no matter how old they are; TRUE means that the client asks for products updated/created within the last REQ\_IF\_MODIFIED\_SINCE minutes.

#### **REQ\_IF\_MODIFIED\_SINCE**

The number of *minutes* for the fresh-products request. This value only matters if REQ\_IF\_MODIFIED was set to TRUE. Example:

REQ\_IF\_MODIFIED\_SINCE = 10

#### **REQ\_UPDATES**

TRUE/FALSE: TRUE means that the client should not exit after submitting the request and processing server's reply. Rather, the retriever would sleep in the background,

and with the period of `REQ_UPDATES_INTERVAL` minutes, *repeatedly* ask the Metcast server for product updates.

#### **REQ\_UPDATES\_INTERVAL**

The number of *minutes* for requesting product updates. Example:

```
REQ_UPDATES_INTERVAL = 15
```

#### **RETRY\_ON\_ERROR**

The number of times the retriever would try to repeat a request that failed because of a *third-party* error. A request may fail for a number of reasons, some of them are the client's fault (for example, a syntax error in the request message), but some of them are not (e.g., a network connection has timed out or unexpectedly terminated, the server is temporarily unavailable, etc). It is the latter errors that may be remedied by retrying the request, a specified number of times.

The field is not mandatory; if it is omitted, some reasonable default will be used (typically 5 times)

Example:

```
RETRY_ON_ERROR = 5
```

#### **Example**

```
# JMV HHTP transport configuration

SERVER_URL = http://152.80.34.202:80/cgi-bin/server
PROXY_ENABLE = TRUE
PROXY_NAME = 192.16.167.45
PROXY_PORT = 80
AUTH_CREDENTIAL = Basic QWxhZGRpbjpvcGVuIHNlc2FtZQ==
AUTH_ENABLE = true
REQ_IF_MODIFIED = true
REQ_IF_MODIFIED_SINCE = 120
REQ_UPDATES = FALSE
REQ_UPDATES_INTERVAL = 15
RETRY_ON_ERROR = 7
```

### **3.3.1 Authorization and Access Authentication**

We adhere to the HTTP standard, with the following simplification:

If the user specified that they want to authenticate themselves to the server, the retriever module computes the basic authorization credential and submits it with the request, in the `Authorization` header. Appendix D contains code to compute a basic authorization credential.

In any case, if the server replies with 401, (Unauthorized) message, the access has been denied, and that is final. We simply log the whole response (which may contain some explanation why access was denied), and quit.

Just for the reference, here is an example of browser-server interaction, when a Netscape browser tried to GET a resource whose access was limited (by an `.htaccess` file placed by Earl in `/~ravide/clams/` directory).

The client (browser) opens the connection and submits the request:

```
GET /~ravide/clams/ HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/3.0 (Macintosh; I; 68K)
Host: 152.80.34.202:4001
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, */
[An empty line]
```

The server replies:

```
HTTP/1.0 401 Unauthorized
Server: Netscape-Communications/1.12
Date: Tuesday, 11-Feb-97 00:30:41 GMT
WWW-authenticate: basic realm="Earls Clams"
Content-type: text/html
Content-length: 223

<HTML><HEAD><TITLE>Unauthorized</TITLE></HEAD>
<BODY><H1>Unauthorized</H1>
Proper authorization is required for this area. Either your browser
does not perform authorization, or your authorization has failed.
</BODY></HTML>
and closes the connection.
```

The (Netscape) browser displays a dialog asking the user to enter an id and a password. After that, the browser opens the connecton again and resubmits the request, this time including the `Authorization` header:

```
GET /~ravide/clams/ HTTP/1.0
Connection: Keep-Alive
User-Agent: Mozilla/3.0 (Macintosh; I; 68K)
Host: 152.80.34.202:4001
Accept: image/gif, image/x-bitmap, image/jpeg, image/pjpeg, */
Authorization: Basic Y2xhbXM6Y2hvd2Rlcg==
[An empty line]
```

The authorization was invalid, and the server simply closed the connection. Obviously the server was too rude (or obsolete): it should have sent the 401 response again.

## **4 USING THE OMNICAST RETRIEVER TO RETRIEVE DATA**

This section describes two methods to use the Omnicast Retriever to retrieve data for a program. The first, and preferred method, is to write a Java program using the Retriever's APIs to create the request, mailcap, and configuration objects and retrieve and process the requested data before supplying it to the calling program. The second method, which may be used by non-Java programs, is to use the command line interface provided by the retriever stub. This method requires creation of request, mailcap, and configuration files. Retrieved data will be processed and routed as specified in the session's mailcap file.

### **4.1 USING THE OMNICAST RETRIEVER JAVA APIs**

This section describes, in tutorial form, the creation of a special purpose Java program that communicates with the Retriever Service in order to make requests to the Metcast Server. It assumes that you have at least some programming experience in Java. Although there are many other methods of interfacing with the Omnicast Retriever, this section's sole purpose is to explain how to interface with the Retriever Service using the preferred method, which is Java and Java's Remote Method Invocation (RMI) facility.

#### **4.1.1 Prerequisites**

Before beginning:

1. Download and install Metcast onto the machine that you wish to do development on. Authorized users may download Metcast for Windows and Solaris via the following link: <http://www.fnmoc.navy.mil> (DOD level 1 or 2 Access is required). Metcast for HP-UX 10.20 is available via DII-COE.

**Note:** Department of Defense (DOD) staff and authorized contractors may request access authorization from the Fleet Support Office via the following link: <http://www.fnmoc.navy.mil/PUBLIC/ADMIN/request.html>.

2. Download and install the Java Software Developers Kit (SDK) version 1.3 that applies to your platform. Download links for various platforms are provided below:

Windows 98/NT/2000: <http://java.sun.com/j2se/1.3/download-windows.html>

Sun Solaris 2.51/2.6/7: <http://www.java.sun.com/j2se/1.3/download-solaris.html>

HP-UX 10.20/11.00:

[http://www.hp.com/products1/unix/java/java2/sdkrte1\\_3/downloads/index.html](http://www.hp.com/products1/unix/java/java2/sdkrte1_3/downloads/index.html)

**Important Note:** HP-UX v10.20 is no longer a suitable platform for development using the latest versions of the Omnicast Retriever. The latest versions of the Omnicast Retriever were built using JRE v1.3, to take advantage of its multi-threading capabilities. Unfortunately, HP does not support multi-thread processing.

3. Certain environment variables need to be configured on your system before you begin developing software to interface with the Omnicast Retriever. These variables are the PATH and CLASSPATH variables.

The PATH environment variable needs to be set so that the directory that includes the Java binaries is included in the default path. The CLASSPATH environment variable needs to point to four Java archive files as well as the current directory so that the Java runtime can find the necessary libraries that are required to run the Omnicast Retriever.

### ***Windows 98***

In Windows 98, do the following:

1. Bring up the autoexec.bat file with notepad or sysedit. The autoexec.bat file is located at c:\ if c is your boot drive.
2. Add the following lines to the end of the autoexec.bat file to specify the PATH and CLASSPATH environment variables for the JDK:

```
SET PATH=C:\JDK1.3\bin;%PATH%
SET CLASSPATH=. ;C:\JDK1.3\lib\classes.zip
SET CLASSPATH=C:\jmwin\noddsfls\lib\omnicast.jar;%CLASSPATH%
SET CLASSPATH=C:\jmwin\noddsfls\lib\kiwi.jar
```

3. Save the edits you made in the autoexec.bat file.
4. Reboot your computer for the changes to take effect

### ***Windows NT 4.0/2000***

In Windows NT 4.0/2000, do the following:

1. In the control panel, select System.
2. Select the Environment tab and add an environment variable named CLASSPATH; set its variable to the following:  
  
. ;C:\JDK1.3\lib\classes.zip;C:\jmwin\noddsfls\lib\omni  
cast.jar;C:\jmwin\noddsfls\lib\kiwi.jar
3. If you have an existing PATH environment variable, edit it so that C:\JDK1.3\bin; (including the semicolon) is prepended to the current PATH value.

4. Open up a MS-DOS Command Prompt window. From inside the Command Prompt window, change directories to the jmvwin\noddsfls directory using a command like the following:

```
cd \jmvwin\noddsfls
```

Open up two more Command prompt windows and cd to \jmvwin\noddsfls for these MS-DOS sessions also.

### **Solaris & HP-UX 10.20**

If you are using Solaris or HP-UX: (Please refer to the Important Note above, if using HP-UX v10.20)

1. Edit the CLASSPATH and PATH environment variables in your shell's init file (such as .kshrc). Add specifications for class directories and path directories so that the omnicast.jar, kiwi.jar, classes.zip, and the Java bin path are included. Separate the directories for the CLASSPATH and PATH specification with a colon, rather than a semicolon.
2. Log out and then log in again so that your changes take effect. Change directories to the jmvwin/noddsfls directory using a command like the following:

```
cd ~/jmvwin/noddsfls
```

Open up two more X-Terminal windows and cd to jmvwin/noddsfls for these terminal sessions also.

## **4.1.2 Creating a Program That Connects With the Omnicast Retriever**

Before interfacing with the Omnicast Retriever, a program must first establish a connection with the Omnicast Retriever Service. The sample Java program on the next page illustrates how to establish a connection with the Omnicast Retriever Service.

The preferred method to start the Retriever Service is via the omniRetriever. In addition to starting the Retriever Service, the omniRetriever launches the Retriever Monitor GUI, which may be used to monitor the status of a retrieval session. To start the omniRetriever on a Windows platform, navigate to the omniRetriever.exe file that is located within the jmvwin\noddsfls directory and double click on the file name to execute it. The omniRetriever may also be started via the DOS command prompt by entering the following:

```
C:\jmvwin\noddsfls>omniRetriever
```

Upon starting the omniRetriever, a blue and white, cloud icon will appear in your system tray, and a process named RetrieverServic will appear in the processes window of the

Task Manager. Double click on the cloud icon to open the Retriever Monitor GUI. To shut down the Retriever Monitor and Retrieval Service, right click on the cloud icon and select the “shutdown Retriever Service” option.

Copy and save the sample java program listed below as RetrieverSample.java in the c:\jmwwin\noddsfls directory and compiled it. Type the following in the command prompt window:

```
c:\jmwwin\noddsfls>java RetrieverSample
```

You should receive some output from the program that says:

```
Connection to the Retriever Service was successful.
```

```
import java.io.*;
import java.util.*;
import omnicast.retrieverService.*;
import omnicast.util.Environment;
import omnicast.shared.*;
import java.rmi.registry.*;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.NotBoundException;
import java.rmi.RMISecurityManager;
import java.net.MalformedURLException;

/* Sample program that communicates with the Retriever Service */
public class RetrieverSample {

    public static void main(String[] args){
        RetrieverServiceInterface serviceObj; // Handle to Remote Retriever
        String localHost; // Local IP Host
        int rmiPort; // Socket Port for Connection
        String rmiPortStr; // String representation
        String connectURL; // URL for remote Retriever

        // Use omnicast utility library to get RMI port setting from
        // the .rsettings file located in current directory.
        rmiPort = omnicast.util.Environment.getRmiPort();
        rmiPortStr = Integer.toString(rmiPort); // Convert to String
        try {
            // Get the IP host for this computer
            localHost=java.net.InetAddress.getLocalHost().toString();
            // Remove extra garbage in name that Windows OS's might return
            localHost=omnicast.winutil.Util.truncateHostName(localHost);
            // Compose the URL for connecting to the Retriever Service
            // Result will look like "//myHost:1099/RetrieverService"
            connectURL="//"+localHost+":"+rmiPortStr+"/RetrieverService";
            // Make the connection to the Retriever Service
            serviceObj=(RetrieverServiceInterface)Naming.lookup(connectURL);
            // Got here so we must have success
            System.out.println("Connection to the Retriever Service "+"was
                successful");
        }
    }
}
```

### **4.1.3 Creating a Retrieval Session**

The first step in creating a retrieval session is to decide on the session's unique identifiers. In the example below, there are three identifiers that are combined to uniquely identify a retrieval session. These are *AREA NAME*, *DATA DOMAIN*, and *SERVER NAME*. Please note that these names are arbitrary and have no specific meaning for the Retriever Service. These names are used for display purposes in the Retriever Monitor and for accessing the Retriever Session object from the Retriever Service object.

- AREA NAME refers to the name you want to assign for the area of your product request.
- DATA DOMAIN refers to the domain the product request data is in such as gridded, observations, or images.
- SERVER NAME refers to the name you would like to call the server that you attach to.

Add the following code to the sample program (RetrieverSample.java) immediately below the line of code that says:

```
System.out.println("Connection to the Retriever Service  
" + "was successful");
```

```
// Create a Retriever Session that has an AREA NAME of myArea, a  
// DATA DOMAIN of gridded, and a SERVER NAME of FNMOC.  
RetrieverSessionInterface remoteSession;  
remoteSession=serviceObj.createRetrieverSession("myArea", "gridded",  
"FNMOC");
```

### **4.1.4 Configuring Retriever Session Properties**

Before a Retriever Session can be executed, its properties must be configured. This is achieved by defining the Configuration Object. As discussed in [Section 3.3](#), the configuration object is a remote interface object that allows a client program to access a Retrieval Session. In the sample program code, the configuration object (named serviceObj) is created after connection to the Omnicast Retriever Service is established. Click on the link below to view descriptions of the various methods available for the RetrieverSessionInterface objects.

[http://www.metnet.navy.mil/~spawar/Omnicast\\_API\\_Doc/omnicast/retrieverService/RetrieverSessionInterface.html](http://www.metnet.navy.mil/~spawar/Omnicast_API_Doc/omnicast/retrieverService/RetrieverSessionInterface.html)

The table below shows a typical Retriever Session configuration. The Request-If-Modified is enabled and set to 720 seconds. The request Mime Type is set to "text/x-

mbl". Note that the Mime Type property should always be set to this value. The Server URL is set and the session is told to retry up to 5 times should an error occur. Add this session configuration code to the RetrieverSample.java file so that it is located immediately below the code that was added in Section 4.1.3 above.

```
// Configure the Retriever Session's Properties
remoteSession.setNetDownAllowanceEnabled(false);
remoteSession.setReqIfModifiedEnabled(true);
remoteSession.setReqIfModifiedSince(720);
remoteSession.setRequestMimeType("text/x-mbl");
remoteSession.setRetryOnErrorCountEnabled(true);
remoteSession.setRetryOnErrorCount(5);
remoteSession.setReqUpdatesEnabled(true);
remoteSession.setReqUpdatesInterval(300); // Check every 5 minutes
remoteSession.setServerUrl("http://152.80.49.210/cgi-
bin/mcsrvr/rest/server");
remoteSession.setMetcastVersion("RetrieverSample");
```

#### 4.1.4.1 Configuring the Retrieval Session to Use Authentication

In the above block of code the Server URL is set to a Metcast server that is restricted. This means that we must give an authorization credential in order to communicate with it. Because this document is not allowed to include the actual authorization credential, you will need to find it yourself and then insert it below where it says "Insert Authorization Credential Here." Authorized users who have obtained a Metcast account from the Fleet Support Office (as described in Section 4.1.1) can find the value for the authorization credential by navigating to the jmvwin\noddsfls\channels directory and looking at the contents of any file with a .con extension.

This file contains an AUTH\_ID value that resembles the one shown below.

```
AUTH_ID = Basic dIWkIk1HXiOOikOw=
```

Set the authCredential string below to the value that you found in the .con file. For the above example, it would be the following:

```
String authCredential="Basic dIWkIk1HXiOOikOw=";
```

Append the following code to RetrieverSample.java. It should be located immediately below the block of code added in the previous section.

```
// Configure to use Authentication
String authCredential="Insert Authorization Credential Here";
remoteSession.setAuthorizationEnabled(true);
remoteSession.setAuthCredential(authCredential);
```

Refer to Appendix D for code that will compute an HTTP 1.1 Basic Authorization Credential. The authorization credential will resemble the following:

```
String authCredential="Basic" + compute_cookie(<username>, <password>);
```

where <username> and <password> are substitutions for an actual username and password.

If utilized, the code in Appendix D should be appended to the RetrieverSample.java program after first adding code to configure the following: a Proxy Server (if necessary), to set the request message, to configure the mailcap object, and to start the retrieval session.

#### **4.1.4.2 Configuring the Retrieval Session to use a Proxy Server**

The Retrieval Session can be configured to use a Proxy server by using the following RetrieverSessionInterface methods: `setProxyEnabled()`, `setProxyPort()`, and `setProxyName()`. Click on the link below to view details for these methods.

[http://www.metnet.navy.mil/~spawar/Omnicast\\_API\\_Doc/omnicast/retrieverService/RetrieverSessionInterface.html](http://www.metnet.navy.mil/~spawar/Omnicast_API_Doc/omnicast/retrieverService/RetrieverSessionInterface.html)

Add the following lines to the sample program below the lines that say:

```
remoteSession.setAuthCredential(authCredential);

// Set the Server Proxy Settings
remoteSession.setProxyEnabled(true);           // Set to false if you don't have a proxy
remoteSession.setProxyName("192.16.167.45"); // Replace with your proxy' IP address here
remoteSession.setProxyPort(80);
// remoteSession.setProxyAuth("Basic Insert Proxy authorization code here"); // Proxy
Authorization, if any
```

If proxy authentication is required, enter your proxy IP address in place of the existing example proxy IP address in the third line above. Next, uncomment the fifth line above and enter the proxy authorization code in place of the text “Insert Proxy authorization code here”

Save and compile the file. With the Omnicast Retriever Service running (see above), run the program by typing the following in a command prompt window:

```
c:\jmwwin\noddsfls>java RetrieverSample
```

You should see the following debug information generated in the Omnicast Retriever command prompt window:

```
Retriever Service: Proxy Enabled set to: true
Retriever Service: Server Proxy Name set to: 192.16.167.45
Retriever Service: Proxy Port set to: 80
```

#### 4.1.4.3 Setting the Request Message for the Retrieval Session

The next step is to set up the request that you will send to the server. See [Section 3.1](#) for information about the request language (MBL). The `setRequestString` method is used to pass the request message to the Omnicast Retriever. The code block below illustrates how to set the request string for a retrieval session.

```
// Define the Request Message String
String requestMessageString =
    "(MYAREA\n"+
    " (bounding-box 57.5 -112.5 7.5 -30.0)\n"+
    " (products\n"+
    "   (grib\n"+
    "     (bounding-box 57.0 -112.0 8.0 -32.0)\n"+
    "     (product-GRIB-code 11) (layer height-ft 1000) (tau 12 36) "+
    "     (source 58) (process-id 10) (resolution 1.000))\n"+
    " ))\n";
remoteSession.setRequestString(requestMessageString);
```

#### 4.1.4.4 Configuring the Mailcap Object for the Retrieval Session

In order to let Omnicast Retriever know what to do with the products after they are retrieved, you need to set up a mailcap for the session. Omnicast Retriever provides a Mailcap Interface with various functions to allow you to set up the mailcap object.

The code fragment below gets the mailcap for the session and adds an entry to it. The entry we add tells the Omnicast Retriever to perform the "copy %s swh.grib" operation every time it downloads product of MIME type "application/x-grib." This code should be added to the `SampleProgram.java` file immediately below the block entered in the previous section.

```
// Configure a Mailcap for the retrieval Session
MailcapInterface remoteMailcap;
MailcapEntryInterface remoteMailcapEntry;
MimeOperationInterface remoteMimeOperation;
String operationString = "copy %s swh.grib";
StringTokenizer st;

// Get the Mailcap
remoteMailcap = (MailcapInterface)remoteSession.getMailcap();
// Create a Mailcap Entry in our Mailcap
remoteMailcapEntry = (MailcapEntryInterface)remoteMailcap.createMailcapEntry();
// Set the MIME type for this Mailcap entry
remoteMailcapEntry.setMimeType("application/x-grib");
// An operation will be performed after data of this MIME type is downloaded
remoteMailcapEntry.setOperationEnabled(true);
// No operation will occur before data of this MIME type is downloaded
remoteMailcapEntry.setPreOperationEnabled(false);
// No operation will occur after all data of this MIME type is downloaded
remoteMailcapEntry.setPostOperationEnabled(false);
// Create a MIME Operation object
remoteMimeOperation = (MimeOperationInterface)remoteMailcapEntry.getOperation();
// Enable this MIME operation
remoteMimeOperation.setOperationEnabled(true);
// Set the arguments of this MIME operation by parsing the operationString
st = new StringTokenizer(operationString);
while (st.hasMoreTokens()) {
```

```
} remoteMimeOperation.addArgument(st.nextToken());
```

#### **4.1.5 Starting the Retrieval Session**

The retrieval session is now fully configured and ready to be started. The following block of code will tell Omnicast Retriever to start the session. It should be entered immediately following the code block entered in the previous section.

```
// Starting the Retrieval Session
System.out.println("Starting retrieval session");
remoteSession.startSession();

}catch (Exception e) {
// Exception occurred. Most likely the Retiever Service is not
// running.
System.out.println("Could not connect to Retriever "+
"Service\n"+e.getMessage());
}
```

Appendix C provides the complete code listing for the sample program. Appendix D contains code to compute an HTTP 1.1 Basic Authorization Credential.

## **4.2 RUNNING THE RETRIEVER USING THE COMMAND LINE INTERFACE**

Programs not written in Java can use the command line interface to run the Omnicast Retriever. The following steps are involved:

1. Create a subdirectory under the jmvwin\noddsfls directory to contain the configuration, request, and mailcap files and receive the downloaded products. This subdirectory should have the same name as the area specified in the request file.
2. Create a configuration file that provides the server connection and authentication parameters (see [Section 3.3](#)). This information should be saved as a text file with a .con extension in the directory created in Step 1.
2. Create a mailcap file that tells the Omnicast Retriever what to do with the data following the download (see [Section 3.2](#)). Save this information in a text file (the filename and extension are not critical) in the directory created in Step 1.
3. Create a request file that tells the METCAST Server what data you require (see [Section 3.1](#)). This information should be saved as a text file with a .mbl extension in the directory created in Step 1.

4. Shell out from your program and run the Omnicast Retriever (`c:\jmwwin\noddsfls\retrievr` on Windows machines). The command line interface is:

```
retrievr [s-config-file req-file-mime-type req-file mailcap-file]
```

where

`s-config-file` is the path to the file containing the configuration parameters,

`req-file-mime-type` is the MIME type of the request file (always set to *text/x-mbl*),

`req-file` is the path to the file containing the request message, and

`mailcap-file` is the path to the mailcap file for the retrieval session.

The downloaded data will be routed according to the specifications in the mailcap file for the retrieval session.

A separate Retriever session is needed for each type of data (gridded, observations, satellite imagery).

## **5 FORMATS FOR RETRIEVED DATA**

All data retrieved by METCAST are enclosed in a MIME wrapper, which is used to determine the processing to be performed on the data per the mailcap specification, and is removed by the METCAST Retriever during processing. The sections below deal the format(s) in which each type of data are provided.

### **5.1 DEALING WITH GRIDDED DATA**

Two-dimensional grids are downloaded in the WMO Gridded Binary (GRIB) format, specified in WMO-386, Volume 1.2. GRIB is an efficient vehicle for transmitting large volumes of gridded data to automated centers over high-speed telecommunication lines using modern protocols. By packing information into the GRIB code, messages (or records - the terms are synonymous in this context) can be made more compact than character oriented bulletins, which will produce faster computer-to-computer transmissions. GRIB can equally well serve as a data storage format, generating the same efficiencies relative to information storage and retrieval devices.

Each GRIB record intended for either transmission or storage contains a single parameter with values located at an array of grid points, or represented as a set of spectral coefficients, for a single level (or layer), encoded as a continuous bit stream. Logical divisions of the record are designated as "sections", each of which provides control information and/or data. A GRIB record consists of six sections, two of which are optional:

- (0) Indicator Section
- (1) Product Definition Section (PDS)
- (2) Grid Description Section (GDS) - optional
- (3) Bit Map Section (BMS) - optional
- (4) Binary Data Section (BDS)
- (5) '7777' (ASCII Characters)

The *Guide to WMO Binary Code Forms*, Part 2, contains a detailed description of the GRIB format. This document will not deal with the actual decoding of the GRIB files, except to mention that there are several GRIB decoders available.

A typical mailcap entry for dealing with GRIB data might be:

```
application/x-grib; copy %s  
"%{area}\\"%{ParName}^%{Model}^%{Lab}^%{Title}^%{Land}^%{HiLO}^%{Data}^%  
{Units}^%{LCV}^%{HCV}^%{Intv}^%{CenV}^%{ModType}^%{DataMod}^%{Sort}^%{P
```

```
arId}^%{Layer}^%{Ht}^%{XRes}^%{YRes}^%{CenterID}^%{SubCenterID}^%{Proce  
ssId}^%{GridID}^%{Epoch}^%{TAU_REQUESTED}^%{TAU_DELIVERED}^%.grib"  
<end>process the GRIB files</end>
```

Here the initial portion copies the data to a file with a complex name that includes the area name, parameter name, model name, and other information about the data and ends in a *.grib* extension. Once this is done, the programmer could do further processing by placing commands between the *<end>* and *</end>* tags.

## 5.2 DEALING WITH OBSERVATION DATA

Meteorological and Oceanographic observations, and certain types of bulletins (SIGMETS, JOTS warnings, and Tropical Cyclone Warnings, for example) are transmitted by the METCAST server in an Extensible Markup Language (XML)-based format called Weather Observation Markup Format (OMF). OMF preserves the original text of each observation or bulletin, and also includes information decoded from the observation/bulletin and other metadata concerning the message (). OMF is described in the paragraphs below.

### 5.2.1 OMF Rationale

Weather Observation Markup Format is an application of XML to describe a particular kind of document: *weather observation reports*.

Weather observations are reported in a variety of formats: FM 15-X Ext. METAR, FM 16-X Ext. SPECI, FM 51-X Ext. TAF, etc. These reports constitute AWDS, NWS etc. feeds. Unfortunately, these formats are rather unsuitable for serving observation information requested for a particular area of interest:

- The reports refer to observation stations only by their call-signs or block-ids. A report itself specifies neither the station's location nor its full name. Every entity that interprets the reports is supposed to possess the current copy of a *Master Station library*, which contains such static information about the stations. Although this arrangement is well suited for processing of a massive feed, it is not convenient for a client that is only interested in weather reports for a (relatively) small area. In this case, the description of an observation station ought to be carried along with its report, as an *annotation*.
- The standard AWN, NWS, WMO reports do not bear the complete timestamps: the time of an observation is recorded only in `HHMM` or `DDHHMM` format, omitting the month and the year. This makes the raw reports unsuitable for archival storage, record keeping, trend processing, etc.
- Report consumers are frequently interested in *derived* quantities like cloud ceiling, flight conditions, or humidity. These parameters can be computed from the data in the

report; however, the computation is sometimes not trivial, and often messy. It makes sense therefore for a server to compute the most popular derived parameters and serve them along with the raw report data.

The intent of OMF therefore is annotation of weather reports. The reports should be distributed as they are without any mangling. On the other hand, we would like to mark them up with station information (location, name), and with a few derived parameters. This document is to define the format of this markup.

OMF is an application of XML, and by its virtue, an application of SGML. SGML is used extensively within DoD for documenting of various types of information (military standards, procurement materials, service manuals). OMF brings weather observations into the same fold.

Thus, the design goals of OMF are:

- Mark up (annotate) raw observation reports with additional description and derived, computed quantities.
- The raw report data must not be modified in any way, and should be conveniently extractable (by simply stripping all the tags away).
- OMF must be concise. While providing useful annotations to a client, OMF markup should not impose undue overhead on communication channels.
- It should be possible to extend the markup with additional annotations, without affecting applications that do not use this information.

### **5.2.2 Structure of an OMF Document**

The Observation Markup Format is an application XML. Thus its definition consists of the various XML elements defined in this section. See the references section for information on XML for further details.

The OMF contains the following elements:

- **Reports** - defines a group of weather observation reports
  - **METAR** for a single METAR report
  - **SPECI** for a single SPECI report
  - **UAR** for a combined Rawinsonde and Pibal Observation report
  - **BTSC** for ocean profile data (temperature, salinity, current)

- **SYN** for a surface synoptic report from a land or sea station
- **PIREP** for a single pilot report.
- **Advisories** - defines a collection of weather hazard warnings
  - **SIGMET** - SIGnificant METeorological Information
  - **AIRMET** – AIRman’s METeorological Information
- **Forecasts** - defines a set of weather forecasts
  - **TAF** - Terminal Aerodrome Forecasts
- **Messages** - defines a set of plain-text bulletins.

The following sections define the major elements along with the minor elements that are relevant to them. In each section, XML DTD declarations are provided for precise definition of elements and attributes. The collection of XML DTD declarations found in this specification can be arbitrarily extended to add new elements and attributes for new enhancements.

Some of elements attributes are common. For compactness, they are defined in the following special section.

### 5.2.2.1 Basic Attributes

The following attributes provide basic annotation information. These attributes may (and often must) be present in appropriate elements describing reports.

**Table 5-1. Basic Attributes of an Observation in OMF**

Attribute	Brief Description	Format	Description
<b>TStamp</b>	Time Stamp	unsigned integer	UTC time in seconds since the Epoch, 00:00:00 Jan 1, 1970 UTC. This is the value returned by a POSIX function <code>time(2)</code> . Example: <code>Tstamp='937507702'</code>
<b>TRange</b>	Time Interval	a string of form "aaa, bbb", where aaa and bbb are unsigned integer numbers specifying the beginning and the end timestamps of the interval.	Timestamps are in seconds since the Epoch, 00:00:00 Jan 1, 1970 UTC. These are the values returned by a POSIX function <code>time(2)</code> . Example: <code>Trange='937832400, 937915200'</code>

**Table 5-1. Basic Attributes of an Observation in OMF**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>
<b>LatLon</b>	Specification of a point on the globe	A string of a form "aaa.bbb, cccddd", where aaa.bbb and cccddd are signed floating point numbers	The latitude and longitude, respectively, of a point on the globe, in whole and fractional degrees. The numbers are positive for Northern latitudes and Eastern longitudes, and negative for Southern latitudes and Western longitudes.  The range of the numbers is [-90.0, 90.0] for latitudes, (-180.0, 180.0] for longitudes.  Example: LatLon='32.433, -99.850'
<b>LatLons</b>	Specification of a sequence of points on the globe	a string of a form "lat1, lon1, lat2, lon2, latn, lonn" where each pair (lat1, lon1, etc.) are signed floating point numbers	A sequence of pairs of numbers, each pair giving the latitude and longitude of a single point in the sequence, in whole and fractional degrees.  See the <b>LatLon</b> attribute above for more details.  Example: LatLons='38.420, -111.125, 36.286, -111.492, 36.307, -112.630, 37.700, -113.223, 38.420, -111.125'
<b>BBox</b>	Bounding box, which tells the latitudal and the longitudinal spans of an area of the globe	A string of a form "lat-N, lon-W, lat-S, lon-E", where the lats and lons are signed floating-point numbers, in degrees	Specification of the bounding box for an area of interest. Here lat-N is the latitude of the Northern-most point of the area, lat-S is the latitude of the Southern-most point, lon-W is the longitude of the Western-most point of the area, and lon-E is the Eastern-most longitude.  It is required that lat-N >= lat-S. The left-lon (lon-W) may however be greater than the right-lon (lon-E). For example, a range of longitudes [-170, 170] specifies the entire world but Indonesia. On the other end, the range [170, -170] includes Indonesia only. By the same token, [-10, 10] pertains to a 21-degree longitude strip along the Greenwich meridian, while [10, -10] specifies the whole globe except for that strip.  Example: Bbox='60.0, -120.0, 20.0, -100.0'

**Table 5-1. Basic Attributes of an Observation in OMF**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>
<b>BId</b>	Station identification group	Unsigned integer	WMO Block Station ID, or other identifier for buoy or ship
<b>SName</b>	Call sign and full name of an observing station	A string of the form "cccccc, name", where ccccc are the call letters of the station (ICAO station id: 4 or 5 upper-case letters, may be omitted), name is an arbitrary string describing the station	The observing stations ICAO, aircraft, or ship call sign, plus a plain-text station name (e.g. "KMRY, Monterey CA Airport")  Example: Sname='KYNL, YUMA (MCAS)'
<b>Elev</b>	Elevation	A non-negative integer, or omitted if unknown.	Station elevation relative to sea level, in meters. This attribute may specify a surface elevation of an observation station, or an upper-air elevation for an upper-air report.  Example: Elev='16'

The XML Document Type Definition (DTD) entries for these attributes are shown below.

### XML DTD for Basic Attributes

```
<!ENTITY % TStamp-type "NMTOKEN">
<!ENTITY % TRange-type "CDATA">
<!ENTITY % TStamp "TStamp %TStamp-type; #REQUIRED">
<!ENTITY % TRange "TRange %TRange-type; #REQUIRED">
<!ENTITY % LatLon "LatLon CDATA #REQUIRED">
<!ENTITY % LatLons "LatLons CDATA #REQUIRED">
<!ENTITY % BBox-REQD "BBox CDATA #REQUIRED">
<!ENTITY % BBox-OPT "BBox CDATA #IMPLIED">
<!ENTITY % BId "BId NMTOKEN #REQUIRED">
<!ENTITY % SName "SName CDATA #REQUIRED">
<!ENTITY % Elev "Elev NMTOKEN #IMPLIED">
```

#### 5.2.2.2 The Reports Element

The Reports element is the container for METAR, SPECI, Synoptic (SYN), PIREPS, Upper Air (UAR), and Bathymeterograph (Temperature, Salinity, Current) (BTSC) reports. Its only attribute is the Tstamp attribute specified in Table 5-1.

**Usage Example:**

```
<Reports Tstamp="937507640">

<METAR ...>

</METAR>

. . .

</Reports>
```

**Expanded OMF DTD for the Reports Element:**

```
<!ELEMENT Reports (METAR|SPECI|PIREP|UAR|BTSC|SYN)*>
<!ATTLIST Reports
      Tstamp NMTOKEN #REQUIRED
>
```

**5.2.2.3 The Forecasts Element**

The Forecasts element is a container for Terminal Aerodrome Forecasts (TAFs). Its only attribute is the Tstamp attribute defined in Table 5-1.

**Usage Example:**

```
<Forecasts Tstamp="937507640">

<TAF ...>

</TAF>

. . .

</Forecasts>
```

**Expanded OMF DTD for the Forecasts Element:**

```
<!ELEMENT Forecasts (TAF)*>
<!ATTLIST Forecasts
      Tstamp NMTOKEN #REQUIRED
>
```

**5.2.2.4 The Advisories Element**

The Advisories element is a container for various types of advisories, such as Significant Meteorological Advisories (SIGMETs), Airmen's Meteorological Information (AIRMETs), and miscellaneous warnings, including JOTS warnings. Its only attribute is the Tstamp attribute defined in Table 5-1.

**Usage Example:**

```
<Advisories Tstamp="937507640">  
  <SIGMET ...>  
  </SIGMET>  
  ...  
</Advisories>
```

**Expanded OMF DTD for the Forecasts Element:**

```
<!ELEMENT Advisories (SIGMET|AIRMET|WW)*>  
<!ATTLIST Forecasts  
      Tstamp NMTOKEN #REQUIRED  
>
```

## 5.2.3 OMF Markup for Each Report Type

The subsections that follow specify the full OMF markup for each type of report. For each report type, the expanded OMF DTD is also provided. Appendix B provides the full XML Document Type Definition for OMF.

### 5.2.3.1 OMF Markup for Observations in METAR Format

METAR reports (normally hourly observations) are contained within a <Reports></Reports> container. The METAR element has the following attributes:

**Table 5-2. OMF Attributes for METAR and SPECI Reports**

Attribute	Brief Description	Format	Description	Req'd?
TStamp	Time Stamp	<-----See Table 5-1----->		Yes
LatLon	Station latitude and longitude	<-----See Table 5-1----->		Yes
Bld	Station Identification Group	Unsigned integer	WMO Block Station ID	Yes
SName	Call sign and full name of an observing station	<-----See Table 5-1----->		Yes
Elev	Station elevation	<-----See Table 5-1----->		No
Vis	Visibility	a number of meters, omitted, or a special token "INF"	Horizontal visibility in meters	No

**Table 5-2. OMF Attributes for METAR and SPECI Reports**

Attribute	Brief Description	Format	Description	Req'd?
Ceiling	Ceiling	a number of feet, omitted, or a special token "INF"	Ceiling in feet	No

The body of the METAR element contains the actual METAR report, as shown in the examples below.

**Example METAR Reports:**

Shown below is a <Reports></Reports> container with three METAR reports. Note that the attributes listed in Table 5-2 are contained within the <METAR ...> tag itself, while the body of the report is contained between the <METAR ...> and </METAR> tags.

```
<Reports TStamp="937841512">
<METAR TStamp='937841040' LatLon='37.350, -78.433' BId='724017'
SName='KFXV, FARMVILLE' Elev='125' Vis='16100' Ceiling='INF'>
KFXV 201524Z AUTO 22007KT 10SM CLR 23/16 A3003 RMK AO1</METAR>
<METAR TStamp='937840920' LatLon='39.083, -77.567' BId='724055'
SName='KJYO, LEESBURG/GODFREY' Elev='119' Vis='16100' Ceiling='INF'>
KJYO 201522Z AUTO 20010G14KT 10SM CLR 23/16 A3001 RMK AO1</METAR>
<METAR TStamp='937839600' LatLon='25.050, -77.467' BId='780730'
SName='MYNN, NASSAU INTL AIRPORT' Elev='7' Vis='INF' Ceiling='1800'>
MYNN 201500Z 14010KT 9999 BKN018 BKN100 29/27 A2992</METAR>
</Reports>
```

**Expanded OMF DTD for the METAR element:**

```
<!ELEMENT METAR (#PCDATA)>
<!ATTLIST METAR
    TStamp      NMOKEN      #REQUIRED
    LatLon     CDATA       #REQUIRED
    BId        NMOKEN      #REQUIRED
    SName      CDATA       #REQUIRED
    Elev        NMOKEN      #IMPLIED
    Vis         NMOKEN      #IMPLIED
    Ceiling    NMOKEN      #IMPLIED
    >
```

**5.2.3.2 OMF Markup for SPECI Observations**

Special observations are reported in the SPECI element, which is exactly synonymous in structure to the METAR element described above. The attributes are listed in Table 5-2.

**Example SPECI Reports:**

This example shows a <Reports></Reports> container containing two SPECI observations. Note that the attributes listed in Table 5-2 are contained within the

<SPECI...> tag itself, while the body of the report is contained between the <SPECI ...> and </SPECI> tags.

```
<Reports TStamp="937841512">
<SPECI TStamp='937841040' LatLon='41.783, -80.700' BId='725256'
SName='KHZY, ASHTABULA' Elev='281' Vis='1610' Ceiling='2300'>
KHZY 201524Z AUTO 20004KT 1SM +RA BR FEW003 BKN023 OVC028 15/14 A2991
RMK AO2 P0013 TSNO</SPECI>
<SPECI TStamp='937841040' LatLon='41.933, -72.683' BId='725080'
SName='KBDL, HARTFORD/BRADLEY' Elev='60' Vis='16100' Ceiling='3000'>
KBDL 201524Z 18012G15KT 10SM BKN030 OVC036 21/14 A3005 RMK AO2</SPECI>
</Reports>
```

### **Expanded OMF DTD for the SPECI Element:**

```
<!ELEMENT SPECI (#PCDATA)>
<!ATTLIST SPECI
  Tstamp      NMTOKEN      #REQUIRED
  LatLon     CDATA        #REQUIRED
  BId        NMTOKEN      #REQUIRED
  SName      CDATA        #REQUIRED
  Elev        NMTOKEN      #IMPLIED
  Vis         NMTOKEN      #IMPLIED
  Ceiling    NMTOKEN      #IMPLIED
>
```

#### **5.2.3.3 OMF Markup for Synoptic (SYN) Observations**

Observations in the WMO SYNOP (FM-12), SYNOP SHIP (FM-13), and BUOY (FM-18) formats are contained in OMF in the SYN element. SYN elements are in turn contained within a <Reports></Reports> container. The SYN element is itself a container for the SYID, SCODE, SYG, and SYSEA elements described below.

The attributes for the SYN element are shown in Table 5-3 below.

**Table 5-3. OMF Attributes for the SYN Element**

Attribute	Brief Description	Format	Description	Req'd?
TStamp	Time Stamp	<-----See Table 5-1----->		Yes
LatLon	Station latitude and longitude	<-----See Table 5-1----->		Yes

**Table 5-3. OMF Attributes for the SYN Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
Bld	WMO Block Station Number	String	<p>For a buoy or other observation platform, this id is a combination of a WMO region number, subarea number (per WMO Code Table 0161), and the buoy type and serial number. This information is reported in Section 0 of a synoptic report.</p> <p>If Section 0 contains a call sign rather than a numerical id (as typical with FM 13 SHIP reports), the <code>Bld</code> attribute is computed as <code>itoa(1000009 + hc) % 2^30</code>, where <code>hc</code> is a numerical representation of the call letters considered as a number in radix 36 notation. For example, "0000" hashes to 0, and "zzzz" hashes to 1,679,615. Note this formula makes the <code>Bld</code> attribute a unique numeric identifier for the station.</p>	Yes
SName	Call sign and full name of an observing station		<-----See Table 5-1----->	Yes
Elev	Station elevation		<-----See Table 5-1----->	No
Title	Report title	String	Title defining type of report: AAXX (FM-12), BBXX (FM-13), or ZZYY (FM-18)	Yes
Stype	Station type	String	Type of station: automated (AUTO) or manned (MANN); defaults to MANN	No

**Expanded OMF DTD for the SYN Element:**

```
<!ELEMENT SYN (SYID, SYCODE?, SYG?, SYSEA?)>
<!ATTLIST SYN
    TStamp      NMOKEN          #REQUIRED
    LatLon     CDATA            #REQUIRED
    BId        NMOKEN          #REQUIRED
    SName      CDATA            #REQUIRED
    Elev       NMOKEN          #IMPLIED
    Title      (AAXX|BBXX|ZZYY) #REQUIRED
    SType      (AUTO|MANN) "MANN"
>
```

**5.2.3.3.1 The SYID Element**

The SYID element annotates identification and position data, which constitute Section 0 of the WMO code forms FM-12, -13, and -18. The WS attribute of this element provides information on instruments and measurement procedures that were used to perform the observation. The body of Section 0 of the report is contained between the <SYID...> and </SYID> tags.

**Attributes:**

- WS    Indicator for wind speed units and instrument certification; "0", "1", "3", "4", or omitted, where:  
    0 means wind speed is estimated, in m/s;  
    1 means wind speed is measured, in m/s;  
    3 means wind speed is estimated, in knots;  
    4 means wind speed is measured, in knots

**Expanded OMF DTD for the SYID Element:**

```
<!ELEMENT SYID (#PCDATA)>
<!ATTLIST SYID
    WS (0|1|3|4)    #IMPLIED
>
```

**Usage Example:**

```
<SYID WS='1'>20171 99398 70666</SYID>
```

**5.2.3.3.2 The SYCODE Element**

The SYCODE element marks up those tokens of the original message that are unparsed or cannot be parsed. Currently Sections 4 and 5 of synoptic reports, which contain regional or national-specific codes, are not parsed. Occasionally some of report's data cannot be parsed because they were wrongly encoded or garbled during transmission.

Regardless of the reason, the unparsed parts of a report are thus clearly identified so an application may attempt to extract or salvage whatever information possible. The present

markup format does not correct or modify these unparsed pieces, and does not annotate them any further. The unparsed tokens identified by a SYCODE element always occur *at the very end* of a report. This implies that there may be at most one SYCODE element within a single SYN report.

The SYCODE element has no attributes.

### **OMF DTD for the SYCODE Element:**

```
<!ELEMENT SYCODE (#PCDATA)>
```

#### **Usage Example:**

```
<SYCODE>/3606 10288 20241 40123 51011 91650 22200 00294 10702 70008 333  
91208 555 11064 22067 31632 43408 61649 356064 354063 351059 347058  
350057 346053</SYCODE>
```

#### **5.2.3.3.3 The SYG Element**

The SYG element describes the basic set of meteorological conditions. The body of the element is Section 1 and Section 3 (if present) from the original report. Element's attributes spell out *most* of this information in "plain text". Some of this information (mostly some climatic data, and regional and special codes) is not decoded.

**Table 5-4. OMF Attributes for the SYG Element**

Attribute	Brief Description	Format	Description	Req'd?
T	Air Temperature	positive, zero, or negative number	Air temperature in degrees Celsius	No
TD	Dew point temperature	positive, zero, or negative number	Dew point temperature in degrees Celsius	No
Hum	Relative humidity	non-negative number	Relative humidity in per cent	No
Tmm	Extreme temperatures over the last 24 hours	a string of a form "mmmm, MMMM" or omitted	Minimum and maximum temperatures (degrees Celsius) over the last 24 hours	No
P	Station pressure	positive number	Atmospheric pressure at station level, in hectoPascals	No
P0	Sea level pressure	positive number	Atmospheric pressure at station, reduced to sea level, in hPa	No
Pd	Pressure tendency	String of form "dddd", or omitted	Pressure tendency during the 3 hours preceding the observation	No
Vis	Visibility	Number of meters, omitted, or a special token "INF"	Horizontal visibility in meters	No

**Table 5-4. OMF Attributes for the SYG Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
Ceiling	Ceiling	Number of feet, omitted, or a special token "INF"	Ceiling in feet	No
Wind	Wind speed and direction	String of form "nnn, mm" or omitted	<p>nnn is a true direction <b>from</b> which the wind is blowing, in degrees, or VAR if " the wind is variable, or all directions or unknown or waves confused, direction indeterminate." This is an integer number within [0, 360), with 0 meaning the wind is blowing from true North, 270 stands for the wind blowing from due West. Normally this number has a precision of 10 degrees.</p> <p>mm is the wind speed in meters per second.</p>	No
Wx	Past and present weather conditions and phenomena	String of four digits, "NOSIG", or omitted	See WMO-306, Code tables 4677 and 4561 for the meaning of the four digits. This attribute is coded as "NOSIG" if there is no significant phenomenon to report. The attribute is omitted if not observed or data is not available (see <i>i_x</i> indicator, Code table 1860).	No
Prec	Precipitation amount	String of form "nnn, hh" or " " or omitted	<p>nnn is the amount of precipitation which has fallen during the period preceding the time of observation. The precipitation amount is a non-negative decimal number, in mm.</p> <p>hh is the duration of the period in which the reported precipitation occurred, in whole hours. This attribute is encoded as " " if no precipitation was observed. The attribute is omitted if unknown or not available (see <i>i_R</i> indicator, Code table 1819). Sea stations typically never report precipitation.</p>	No

**Table 5-4. OMF Attributes for the SYG Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
Clouds	Amounts and types of cloud cover	String of five symbols "tplmh" or omitted	The first digit is the total cloud cover in octas (Code table 2700). The second digit is the cloud cover of the lowest clouds, in octas. The other three symbols are types of low, middle, and high clouds, resp. See WMO-306 Code tables for more details.	No

**Usage Example:**

```
<SYG T='14.5' TD='11.7' P0='1015.3' Pd='1 1.1' Wind='310, 1'>46///
/3101 10145 20117 40153 51011 91750</SYG>
```

**OMF DTD for the SYG Element:**

```
<!ELEMENT SYG (#PCDATA)>
<!ATTLIST SYG
    T          NMTOKEN      #IMPLIED
    TD         NMTOKEN      #IMPLIED
    Hum        NMTOKEN      #IMPLIED
    Tmm        CDATA        #IMPLIED
    P          NMTOKEN      #IMPLIED
    P0         NMTOKEN      #IMPLIED
    Pd         NMTOKENS     #IMPLIED
    Vis        NMTOKEN      #IMPLIED
    Ceiling    NMTOKEN      #IMPLIED
    Wind       CDATA        #IMPLIED
    WX         CDATA        #IMPLIED
    Prec       CDATA        #IMPLIED
    Clouds    CDATA        #IMPLIED
>
```

**5.2.3.3.4 The SYSEA Element**

The SYSEA element reports information on sea surface conditions -- in both raw and decoded formats. It is present only when annotating SHIP and BUOY reports. The body of the element is the encoded information about sea conditions exactly as it appears in the raw report, Section 2. A number of attributes present parameters in a fully-decoded, self-contained format. Although all of Section 2 of the original report is carried in a SYSEA element, not all of it is decoded.

**Table 5-5. OMF Attributes for the SYSEA Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
T	Sea surface temperature	Positive, zero, or negative number	Sea surface temperature in degrees Celsius	No
Wave	Sea wave period and height	String of form "pp, hh" or omitted	pp is the period of wind waves in seconds. hh is the height of wind waves, in meters. If a report carries both estimated and measured wind wave data, the instrumented information is preferred.	No
SDir	Ship's course and speed	String of form "nnn, mm" or omitted.	nnn is a true direction of resultant displacement of the ship during the three hours preceding the time of observation. The number is in degrees, or VAR if "variable, or all directions or unknown or waves confused, direction indeterminate." This is an integer number within [0, 360), with 0 meaning the ship has moved towards the true North; 270 means the ship has moved to the West. Normally this number has a precision of 45 degrees.  mm is the average speed made good during the three hours preceding the time of observation, in meters per second.	No

**Usage Example:**

```
<SYSEA T='20.1' Wave='3, 1.5' SDir='90, 7.5'>22223 02201 20303</SYSEA>
```

**OMF DTD for the SYSEA Element:**

```
<!ELEMENT SYSEA (#PCDATA)>
<!ATTLIST SYSEA
    T          NMOKEN      #IMPLIED
    Wave       CDATA        #IMPLIED
    SDdir      CDATA        #IMPLIED
    >
```

### 5.2.3.3.5 Synoptic Report Examples

This example shows two synoptic reports, one from a buoy (title ZZYY) and one from a ship (title BBXX).

```
<Reports TStamp="937854711">
<SYN Title='ZZYY' TStamp='937839600' LatLon='42.359, -51.762'
BId='44627'
SName=, ' SType='AUTO'>
<SYID>20099 1500/ 742359 051762 6112/</SYID>
<SYCODE>444 2012/ 20099 1215/</SYCODE>
<SYG P0='1027.2' Pd='2 0.9' Clouds='0///'>11119 0/// 30272 40272
52009</SYG>
<SYSEA T='20.0'>22219 00200</SYSEA>
</SYN>
<SYN Title='BBXX' TStamp='937850400' LatLon='33.600, -71.700'
BId='2186329'
SName='PFDC, ' SType='MANN'>
<SYID WS='0'>20180 99336 70717</SYID>
<SYCODE>310// 41008 80210</SYCODE>
<SYG T='26.1' TD='18.6' P0='1014.8' Pd='7 -2.5' Vis='20000'
Ceiling='900' Wind='60, 8' WX='0211' Clouds='52252'>41498 50608 10261
20186 40148 57025 70211 82252</SYG>
<SYSEA T='26.0' Wave='4, 1.5' SDir='225, 7.5'>22253 00260 20403</SYSEA>
</SYN>
</Reports>
```

### 5.2.3.4 OMF Markup for Pilot Weather Reports (PIREPS)

The PIREP tag defines a single Pilot Weather Report from an individual aircraft. PIREP reports are wrapped within a <Reports></Reports> container, and also enclose the PIID, PIC, and PIRMKs Elements. Table 5-6 below, describes the seven attributes of the PIREP element.

**Table 5-6. OMF Attributes for the PIREP Reports**

Attribute	Brief Description	Format	Description	Req'd?
Rtype	Report Type	UA or UUA	Routine or Urgent	Yes
TStamp	Time Stamp	<-----See Table 5-1----->		Yes
LatLon	Station latitude and longitude	<-----See Table 5-1----->		Yes
Bld	Station Identification Group	Unsigned integer	WMO Block Station ID	Yes
SName	Call sign of the VHF navaid or airport specified in report	<-----See Table 5-1----->		Yes

**Table 5-6. OMF Attributes for the PIREP Reports**

Attribute	Brief Description	Format	Description	Req'd?
WMO Header	WMO header of bulletin containing the PIREP	Example: UAXX1 KAWN 091301 RTD05		Yes
FLevel	Altitude/Flight level	value in hundreds of feet, omitted, if unknown.	Altitude/Flight Level above Mean Sea Level	No

**Expanded OMF DTD for the PIREP Element:**

```
<!ELEMENT PIREP(PIID, PIC, PIRMK)>
<!ATTLIST PIREP
    Rtype (ROUTINE | URGENT)      #REQUIRED
    TStamp          NMTOKEN        #REQUIRED
    LatLon         CDATA          #REQUIRED
    Bid            NMTOKEN        #REQUIRED
    SName          CDATA          #REQUIRED
    WMOHeader      NMTOKEN        #REQUIRED
    Flevel         NMTOKEN        #IMPLIED
>
```

**Usage Example:**

```
<PIREP Rtype='ROUTINE' TStamp='937857300' LatLon='69.95, 17.2667'
      Bid='434310' Sname='TOL' WMOHeader='UAXX1 KAWN 091301 RTD05'
      Flevel='30'>
.
.
.
</PIREP>
```

**5.2.3.4.1 The PIID Element:**

The PIID element annotates identification, time, and position data, which constitute the following sections of a PIREP: /OV, /TM, and /FL. The element is mandatory and contains a single attribute, AType (/TP). This attribute presents the type of aircraft from which a report originates. This attribute is omitted if the aircraft type is unknown.

**Expanded OMF DTD for the PIID Element**

```
<!ELEMENT PIID  (#PCDATA) >
<!ATTLIST PIID
    AType  CDATA      #IMPLIED
>
```

**Usage Example:**

```
<PIID AType='BE20'>
    MRY UA /OV MRY180010/TM 1545/FL 130/TP BE20
</PIID>
```

#### **5.2.3.4.2 The PIC Element:**

The PIC element describes a set of the meteorological conditions experienced by the pilot at the time of the report. Specifically, it annotates the air temperature and the wind vector (direction and speed), which are represented by the following sections of a PIREP: /TA and /WV. If the PIREP contains neither the air temperature nor the wind data, the PIC element is omitted.

**Table 5-7. OMF Attributes for the PIC Element**

Attribute	Brief Description	Format	Description	Req'd?
T	Air Temperature	positive, zero or negative number in degrees Celsius	Air Temperature in degrees Celsius at the reported Flight Level. If below zero, prefixed with an "M".	No
Wind	Wind vector	String of form "direction, speed" or omitted	Wind direction from which the wind is blowing, coded in tens of degrees using three digits. If direction is less than 100 degrees, direction shall be preceded by a zero. Wind speed in whole knots, coded as a two or three digit group using the hundreds and/or tens digits.	No

#### **Expanded OMF DTD for the PIC Element**

```
<!Element PIC (#PCDATA)>
<!ATTLIST PIC
    T      NMOKEN    #IMPLIED
    Wind   CDATA      #IMPLIED
>
```

#### **Usage Example:**

```
<PIC T='20' Wind='090, 45'</PIC>
```

#### **5.2.3.4.3 The PIRMKS Element**

The PIRMKS element constitutes the body of a PIREP that presents remarks about any combination of the following meteorological phenomena significant to a pilot during the aircraft's flight: flight visibility and weather, turbulence, sky condition, icing, and general remarks. This information is annotated with the help of elements internal to the PIRMKS element. These elements are PIWX, PISKY, PITB, PIICE, and PIRMKS which annotate the following sections of a PIREP respectively: /WX, /SK, /TB, /IC, and /RM. The PIRMKS elements and all of its internal elements are optional.

## **Expanded OMF DTD for the PIRMKS Element**

```
<!ELEMENT PIRMKS ( PIWX?, PISKY?, PITB?, PIICE?, PIRMK? )>
```

### **PIWX Element**

The PIWX element annotates the flight visibility and flight weather information presented in the report. Specifically, the element annotates the following section of the PIREP report: /WX. If the PIREP report does not contain this section, the PIWX element is omitted. The element does not contain any attributes and can only appear within the PIRMKS element.

```
<!ELEMENT PIWX (#PCDATA)>
```

### **PITB Element**

The PITB element annotates the turbulence information presented in a PIREP report. Specifically, the element annotates the following section of a PIREP report: /TB. If the PIREP report does not contain this section, the PITB element is omitted. The element does not contain any attributes and can only appear within the PIRMKS element.

```
<!ELEMENT PITB (#PCDATA)>
```

### **PISKY Element**

The PISKY element annotates the sky condition information presented in a PIREP report. Specifically, the element annotates the following section of a PIREP report: /SK. If the PIREP report does not contain this section, the PISKY element is omitted. The element does not contain any attributes and can only appear within the PIRMKS element.

```
<!ELEMENT PISKY (#PCDATA)>
```

### **PIICE Element**

The PIICE element annotates the icing information presented in a PIREP report. Specifically, the element annotates the following section of a PIREP report: /IC. If the PIREP report does not contain this section, the PIICE element is omitted. The element does not contain any attributes and can only appear within the PIRMKS element.

```
<!ELEMENT PIICE (#PCDATA)>
```

## **PIRMK Element**

The PIRMK element annotates information regarding a phenomenon that is considered important, but does not fit in any other sections of a PIREP report. Specifically, the element annotates the following section of a PIREP report: /RM. If the PIREP report does not contain this section, the PIRMK element is omitted. The element does not contain any attributes and can only appear within the PIRMK element.

### **Usage Example of the PIRMKS Element**

```
<PIRMKS>
    <PIWX>/WX FV05SM -RA</PIWX>
    <PITB>/TB MOD</PITB>
    <PISKY>/SK SCT030-TOP060/OVC120</PISKY>
    <PIICE>/IC MOD CLR 035-055</PIICE>
    <PIRMK>/RM LLWS -15 KT SFC-030 DURC RWY22R SFO</PIRMK>
```

#### **5.2.3.4.4 Examples of complete PIREPs**

```
PIREP RType='ROUTINE' TStamp='950104000' LatLon='49.95, 7.2667'
    BId='474220' SName='MYV' WMOHeader='UAXX1 KAWN 091300 RTD05'
    FLevel='9'>
        <PIID AType='C177'>
            MYV UA /OV MYV120015/TM 1326/FL009/TP C177
        </PIID>
        <PIRMKS>
            <PITB>/TB MOD SFC-009</PITB>
            <PIRMK>/RM DURGC</PIRMK>
        </PIRMKS>
    </PIREP>

<PIREP RType='ROUTINE' TStamp='950105840' LatLon='59.95, 17.2667'
    BId='574020' SName='OTM' WMOHeader='UAXX1 KAWN 091300 RTD05'
    FLevel='160'>
        <PIID AType='BE20'>
            OTM UA /OV OTM090020/TM 1325/FL160/TP BE20
        </PIID>
        <PIRMKS>
            <PIICE>/IC MDT RIME</PIICE>
        </PIRMKS>
    </PIREP>

<PIREP RType='ROUTINE' TStamp='950104980' LatLon='69.95, 27.2667'
    BId='674430' SName='VIS' WMOHeader='UAXX1 KAWN 091300 RTD05'
    FLevel='80'>
        <PIID AType='BE35'>
            VIS UA /OV VIS/TM 1847/FL080/TP BE35
        </PIID>
        <PIRMKS>
            <PIWX>/WX FV10</PIWX>
            <PITB>/TB LGT</PITB>
            <PIRMK>/RM HAZE</PIRMK>
        </PIRMKS>
    </PIREP>
```

### 5.2.3.5 OMF Markup for Upper Air Reports

The UAR element defines one particular *combined* Rawinsonde and Pibal observation report. Its main contents are a sequence of UALEVEL elements that describe a set of weather conditions at particular vertical atmospheric pressure levels.

Raw rawinsonde observation reports are highly unsuitable for any further processing or even human analysis. Not only is the information spread through several parts, which are delivered in separate messages; the information is also encoded; without an extensive set of code tables (Appendix E-III of FMH-3) the data are inaccessible. The OMF markup aims to aid processing and understanding of a report by annotating the raw data with decoded quantities (pressure, temperature, dew point depression, wind vector) in conventional and self-evident units. Furthermore, a reference to an observation station is annotated with the station's full name, location and elevation; time stamps are always presented in EPOCH seconds.

The OMF markup described in this document only annotates the raw text of Rawinsonde and Pibal Observation reports, without altering them or omitting a single character. An annotated UAR report combines all four possible parts of the original report (as well as accompanying Pibal reports if applicable) and presents the levels in a logical sequence. The markup contains explicit "pointers" that refer back to originating parts. Thus it is *always* possible to reconstruct the original messages of a raw report. In case of a coding or transmission error, the corresponding data are marked with a specific tag. It is up to an application then to try to correct the error and salvage whatever data it can.

The UAR element is a container for UAPART, UAID, UACODE, and UALEVELS elements, and is itself contained within a <Reports></Reports> container.

The UAR element takes the following attributes, all specified in Table 5-1:

- TStamp              Time Stamp (epoch seconds)
- LatLon              Latitude and longitude of reporting station
- BId                WMO Block Station Number
- SName              Station name
- Elev                Station elevation. In cases where the Master Station List contains more than one elevation for a station, this attribute will contain the Upper Air elevation for UAR reports.

All of the elements except Elev are required.

#### Expanded OMF DTD for the UAR Element:

```
<!ELEMENT UAR (UAPART+, UAID*, UACODE*, UALEVELS)>

<!ATTLIST UAR
  Tstamp      NMTOKEN      #REQUIRED
  LatLon     CDATA        #REQUIRED
  BId        NMTOKEN      #REQUIRED
  SName      CDATA        #REQUIRED
  Elev       NMTOKEN      #IMPLIED
>
```

#### **5.2.3.5.1   *The UAPART Element***

The UAPART element identifies a part of a combined Rawinsonde/Pibal observation. The results of an upper-air observation are reported in up to four separate messages; a rawinsonde report may also be accompanied by related pilot balloon reports (which may include four parts as well). Each part included in the merged UAR will have a separate UAPART; there may be many UAPARTs for a given sounding.

The body of the element contains the YYGGI<sub>d</sub> group from Section 1 of Part A or Part C of a rawinsonde report, or the YYGGA<sub>4</sub> group from Section 1 of of Part B or Part D of a rawinsonde report or all Parts of a PILOT report. The YYGGI<sub>d</sub> portion of the group identifies the day of the month (YY) and hour of the day (GMT) (GG) of the sounding. The I<sub>d</sub> parameter identifies the hundreds (Part A) or tens (Part C) of hectopascals of pressure relative to the last standard isobaric surface for which wind is reported (WMO-306, Code Table 1734). The a<sub>4</sub> parameter identifies the type of measuring equipment used for winds (WMO-306, Code Table 0265).

The only parameter for the UAPART element is id, which identifies the report type. Values are:

- TTAA, TTBB, TTCC, TTDD for Parts A, B, C, and D, respectively, of a TEMP (FM-35) report,
- UUAA, UUBB, UUCC, UUDD for Parts A, B, C, and D, respectively, of a TEMP SHIP (FM-36) report,
- PPAA, PPBB, PPCC, and PPDD for Parts A, B, C, and D, respectively, of a PILOT (FM-32) report, and
- QQAA, QQBB, QQCC, and QQDD for Parts A, B, C, and D, respectively of a PILOT SHIP (FM-33) report.

#### **Usage Examples:**

The following contains the UAPARTs for a single sounding:

```
<UAPART id='TTAA'>7018/</UAPART>
<UAPART id='TTAA'>7018/</UAPART>
<UAPART id='TTAA'>7018/</UAPART>
```

```
<UAPART id='TTBB'>70180</UAPART>
<UAPART id='TTBB'>70180</UAPART>
<UAPART id='TTCC'>7018</UAPART>
<UAPART id='TTCC'>7018</UAPART>
<UAPART id='TTDD'>7018</UAPART>
<UAPART id='TTDD'>7018</UAPART>
```

### **OMF DTD for the UAPART Element:**

```
<!ELEMENT UAPART (#PCDATA)>

<!ATTLIST UAPART
      id          NMTOKEN          #REQUIRED
      <
```

#### **5.2.3.5.2   *The UAID Element***

The UAID element provides more details about conditions and circumstances under which a particular rawinsonde observation was carried out. Specifically, this element provides data on the particular sounding and tracking system used, precise time of the observation, and, optionally, the sea-surface temperature. The body of the UAID element is two or three words that constitute Section 7 of a report (the words that follow the section identifier 31313).

The only parameter for the UAID element is Ref, a reference identifier specific to the part of the message from which this UAID element was taken. This same Ref is also used in the UACODE and UALEVELS elements, and provides a way to relate these elements to the message part from which they came.

#### **Usage Example:**

```
<UAID Ref='TTBB'>45202 81723</UAID>
```

### **OMF DTD for the UAID Element:**

```
<!ELEMENT UAID (#PCDATA)>
<!ATTLIST UAID
      Ref          NMTOKEN          #REQUIRED
      >
```

#### **5.2.3.5.3   *The UACODE Element***

The UACODE element marks up those tokens of the original message that are unparsed or cannot be parsed. Currently Sections 8, 9 and 10 of a rawinsonde report, and Sections 4, 5 and 6 of a pibal report are not parsed. These sections either contain regional codes, or specify levels that do not have any attributed pressure value. See Appendices E-I and E-II of FMH-3 for more details.

Occasionally some of a report's data cannot be parsed because they were wrongly encoded or garbled during transmission.

Regardless of the reason, the unparsed parts of a report are preserved in the UACODE element and clearly identified so an application may attempt to extract or salvage whatever information possible. The present markup format does not correct or modify these unparsed pieces, and does not annotate them any further. The unparsed tokens identified by an UACODE element always occur *at the very end* of a part referred to by a Ref attribute. This implies that there may be at most one UACODE element referring to a particular part. As a UAR element however may include several UAPARTs, there may be several UACODE elements within a single UAR report.

The UACODE element's only attribute is Ref, which provides a reference to the part of the report from which the unparsed data were taken.

### **Usage Examples:**

```
<UACODE Ref='TTAA'>51515 10142 10194 ///// /////</UACODE>
<UACODE Ref='TTBB'>41414 00900 51515 10142 10150</UACODE>
<UACODE Ref='TTCC'>88999</UACODE>
<UACODE Ref='TTDD'>51515 10150</UACODE>
```

### **OMF DTD for the UACODE Element:**

```
<!ELEMENT UACODE (#PCDATA)>
<!ATTLIST UACODE
      Ref          NMTOKEN          #REQUIRED
      >
```

#### **5.2.3.5.4 The UALEVELS Element**

This element contains a sequence of UALEVEL elements, each of which contains data for a particular vertical atmospheric level. The sequence of UALEVELs is ordered in descending order with respect to the levels' pressures. Certain pressure levels -- for example, surface, tropopause, or maximum wind -- can be described in several parts of the original observation report. Therefore, there may be several UALEVELs corresponding to exactly the same pressure value. The other attributes of these levels may be identical as well, or may differ. It is up to an application to choose the most appropriate values.

The UALEVELS element has no attributes, but merely serves as a container for UALEVEL elements.

### **Usage Example:**

This example shows a UALEVELS container holding all UALEVEL elements for a single sounding.

```
<UALEVELS>
<UALEVEL Ref='TTAA' P='1008.0' H='SURF' T='26.2' DP='4.3' Wind='130,
2'>99008 26243 13003</UALEVEL>
<UALEVEL Ref='TTAA' P='1008.0' H='SURF' T='26.2' DP='4.3' Wind='130,
2'>99008 26243 13003</UALEVEL>
```

<UALEVEL Ref='TTAA' P='1008.0' H='SURF' T='26.2' DP='4.3' Wind='130, 2'>99008 26243 13003</UALEVEL>  
<UALEVEL Ref='TTBB' P='1008.0' H='SURF' T='26.2' DP='4.3'>00008 26243</UALEVEL>  
<UALEVEL Ref='TTBB' P='1008.0' H='SURF' T='26.2' DP='4.3'>00008 26243</UALEVEL>  
<UALEVEL Ref='TTAA' P='1000.0' H='83' T='25.6' DP='4.1' Wind='130, 4'>00083 25641 13007</UALEVEL>  
<UALEVEL Ref='TTAA' P='1000.0' H='83' T='25.6' DP='4.1' Wind='130, 4'>00083 25641 13007</UALEVEL>  
<UALEVEL Ref='TTAA' P='1000.0' H='83' T='25.6' DP='4.1' Wind='130, 4'>00083 25641 13007</UALEVEL>  
<UALEVEL Ref='TTBB' P='947.0' T='21.4' DP='1.5'>11947 21415</UALEVEL>  
<UALEVEL Ref='TTBB' P='947.0' T='21.4' DP='1.5'>11947 21415</UALEVEL>  
<UALEVEL Ref='TTBB' P='933.0' T='22.6' DP='5.0'>22933 22650</UALEVEL>  
<UALEVEL Ref='TTBB' P='933.0' T='22.6' DP='5.0'>22933 22650</UALEVEL>  
<UALEVEL Ref='TTAA' P='925.0' H='765' T='22.2' DP='5.0' Wind='145, 14'>92765 22250 14527</UALEVEL>  
<UALEVEL Ref='TTAA' P='925.0' H='765' T='22.2' DP='5.0' Wind='145, 14'>92765 22250 14527</UALEVEL>  
<UALEVEL Ref='TTAA' P='925.0' H='765' T='22.2' DP='5.0' Wind='145, 14'>92765 22250 14527</UALEVEL>  
<UALEVEL Ref='TTAA' P='850.0' H='1496' T='17.2' DP='2.2' Wind='165, 12'>85496 17222 16525</UALEVEL>  
<UALEVEL Ref='TTAA' P='850.0' H='1496' T='17.2' DP='2.2' Wind='165, 12'>85496 17222 16525</UALEVEL>  
<UALEVEL Ref='TTAA' P='850.0' H='1496' T='17.2' DP='2.2' Wind='165, 12'>85496 17222 16525</UALEVEL>  
<UALEVEL Ref='TTBB' P='820.0' T='15.0' DP='1.0'>33820 15010</UALEVEL>  
<UALEVEL Ref='TTBB' P='820.0' T='15.0' DP='1.0'>33820 15010</UALEVEL>  
<UALEVEL Ref='TTAA' P='700.0' H='3129' T='7.2' DP='0.8' Wind='175, 13'>70129 07208 17526</UALEVEL>  
<UALEVEL Ref='TTAA' P='700.0' H='3129' T='7.2' DP='0.8' Wind='175, 13'>70129 07208 17526</UALEVEL>  
<UALEVEL Ref='TTAA' P='700.0' H='3129' T='7.2' DP='0.8' Wind='175, 13'>70129 07208 17526</UALEVEL>  
<UALEVEL Ref='TTBB' P='700.0' T='7.2' DP='0.8'>44700 07208</UALEVEL>  
<UALEVEL Ref='TTBB' P='700.0' T='7.2' DP='0.8'>44700 07208</UALEVEL>  
<UALEVEL Ref='TTBB' P='682.0' T='6.8' DP='3.7'>55682 06837</UALEVEL>  
<UALEVEL Ref='TTBB' P='682.0' T='6.8' DP='3.7'>55682 06837</UALEVEL>  
<UALEVEL Ref='TTBB' P='657.0' T='6.2' DP='3.7'>66657 06237</UALEVEL>  
<UALEVEL Ref='TTBB' P='657.0' T='6.2' DP='3.7'>66657 06237</UALEVEL>  
<UALEVEL Ref='TTBB' P='590.0' T='1.4' DP='3.2'>77590 01432</UALEVEL>  
<UALEVEL Ref='TTBB' P='590.0' T='1.4' DP='3.2'>77590 01432</UALEVEL>  
<UALEVEL Ref='TTBB' P='553.0' T='-1.4' DP='0.4'>88553 01504</UALEVEL>  
<UALEVEL Ref='TTBB' P='553.0' T='-1.4' DP='0.4'>88553 01504</UALEVEL>  
<UALEVEL Ref='TTAA' P='500.0' H='5840' T='-4.8' DP='1.0' Wind='240, 10'>50584 04910 24021</UALEVEL>  
<UALEVEL Ref='TTAA' P='500.0' H='5840' T='-4.8' DP='1.0' Wind='240, 10'>50584 04910 24021</UALEVEL>  
<UALEVEL Ref='TTAA' P='500.0' H='5840' T='-4.8' DP='1.0' Wind='240, 10'>50584 04910 24021</UALEVEL>  
<UALEVEL Ref='TTBB' P='470.0' T='-7.2' DP='1.0'>99470 07310</UALEVEL>  
<UALEVEL Ref='TTBB' P='470.0' T='-7.2' DP='1.0'>99470 07310</UALEVEL>  
<UALEVEL Ref='TTBB' P='415.0' T='-12.2' DP='4.3'>11415 12343</UALEVEL>  
<UALEVEL Ref='TTBB' P='415.0' T='-12.2' DP='4.3'>11415 12343</UALEVEL>

```
<UALEVEL Ref='TTAA' P='400.0' H='7570' T='-14.6' DP='7.0' Wind='355,
8'>40757 14757 35515</UALEVEL>
<UALEVEL Ref='TTAA' P='400.0' H='7570' T='-14.6' DP='7.0' Wind='355,
8'>40757 14757 35515</UALEVEL>
<UALEVEL Ref='TTAA' P='400.0' H='7570' T='-14.6' DP='7.0' Wind='355,
8'>40757 14757 35515</UALEVEL>
<UALEVEL Ref='TTBB' P='383.0' T='-17.0' DP='9.0'>22383 17159</UALEVEL>
<UALEVEL Ref='TTBB' P='383.0' T='-17.0' DP='9.0'>22383 17159</UALEVEL>
<UALEVEL Ref='TTBB' P='347.0' T='-23.6' DP='4.7'>33347 23747</UALEVEL>
<UALEVEL Ref='TTBB' P='347.0' T='-23.6' DP='4.7'>33347 23747</UALEVEL>
<UALEVEL Ref='TTBB' P='312.0' T='-26.8' DP='3.8'>44312 26938</UALEVEL>
<UALEVEL Ref='TTBB' P='312.0' T='-26.8' DP='3.8'>44312 26938</UALEVEL>
<UALEVEL Ref='TTAA' P='300.0' H='9680' T='-28.8' DP='4.0' Wind='240,
8'>30968 28940 24016</UALEVEL>
<UALEVEL Ref='TTAA' P='300.0' H='9680' T='-28.8' DP='4.0' Wind='240,
8'>30968 28940 24016</UALEVEL>
<UALEVEL Ref='TTAA' P='300.0' H='9680' T='-28.8' DP='4.0' Wind='240,
8'>30968 28940 24016</UALEVEL>
<UALEVEL Ref='TTAA' P='250.0' H='10960' T='-38.6' DP='4.5' Wind='230,
20'>25096 38745 23039</UALEVEL>
<UALEVEL Ref='TTAA' P='250.0' H='10960' T='-38.6' DP='4.5' Wind='230,
20'>25096 38745 23039</UALEVEL>
<UALEVEL Ref='TTAA' P='250.0' H='10960' T='-38.6' DP='4.5' Wind='230,
20'>25096 38745 23039</UALEVEL>
<UALEVEL Ref='TTBB' P='250.0' T='-38.6' DP='4.5'>55250 38745</UALEVEL>
<UALEVEL Ref='TTBB' P='250.0' T='-38.6' DP='4.5'>55250 38745</UALEVEL>
<UALEVEL Ref='TTAA' P='200.0' H='12450' T='-51.4' DP='5.0' Wind='230,
22'>20245 51550 23044</UALEVEL>
<UALEVEL Ref='TTAA' P='200.0' H='12450' T='-51.4' DP='5.0' Wind='230,
22'>20245 51550 23044</UALEVEL>
<UALEVEL Ref='TTAA' P='200.0' H='12450' T='-51.4' DP='5.0' Wind='230,
22'>20245 51550 23044</UALEVEL>
<UALEVEL Ref='TTAA' P='150.0' H='14250' T='-68.0' DP='5.0' Wind='255,
20'>15425 68150 25539</UALEVEL>
<UALEVEL Ref='TTAA' P='150.0' H='14250' T='-68.0' DP='5.0' Wind='255,
20'>15425 68150 25539</UALEVEL>
<UALEVEL Ref='TTAA' P='150.0' H='14250' T='-68.0' DP='5.0' Wind='255,
20'>15425 68150 25539</UALEVEL>
<UALEVEL Ref='TTBB' P='141.0' T='-70.8' DP='4.9'>66141 70949</UALEVEL>
<UALEVEL Ref='TTBB' P='141.0' T='-70.8' DP='4.9'>66141 70949</UALEVEL>
<UALEVEL Ref='TTBB' P='127.0' T='-70.2' DP='6.0'>77127 70356</UALEVEL>
<UALEVEL Ref='TTBB' P='127.0' T='-70.2' DP='6.0'>77127 70356</UALEVEL>
<UALEVEL Ref='TTBB' P='105.0' T='-78.6' DP='6.0'>88105 78756</UALEVEL>
<UALEVEL Ref='TTBB' P='105.0' T='-78.6' DP='6.0'>88105 78756</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>10662 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='TROP' T='-80.4' DP='6.0' Wind='280,
8'>88100 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='TROP' T='-80.4' DP='6.0' Wind='280,
8'>88100 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>10662 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>10662 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='TROP' T='-80.4' DP='6.0' Wind='280,
8'>88100 80556 28015</UALEVEL>
<UALEVEL Ref='TTBB' P='100.0' T='-80.4' DP='6.0'>99100 80556</UALEVEL>
```

```
<UALEVEL Ref='TTBB' P='100.0' T='-80.4' DP='6.0'>99100 80556</UALEVEL>
</UALEVELS>
```

### **OMF DTD for the UALEVELS Element:**

```
<!ELEMENT UALEVELS (UALEVEL*)>
```

#### **5.2.3.5.5 The UALEVEL Element**

The UALEVEL element conveys detailed information about one particular vertical level of the sounding (an isobaric level, surface, tropopause, level of maximum wind, etc.) - in both raw and decoded formats. The body of the element is encoded information about a level exactly as it appeared in one particular part of a report. A required P attribute tells the pressure value of a UALEVEL element; a required Ref element points to a part of the original rawinsonde or pibal message that carried these encoded data. Together, P and Ref attributes identify a UALEVEL. As mentioned above, one P attribute may not be enough as some pressure levels may be described in several parts of a report - in identical *or* different terms. To uniquely identify a UALEVEL one also has to heed a H attribute: For example, if the maximum wind level happens to coincide with one of the standard pressure levels in Part TTAA, there will be two UALEVEL elements with identical values of P and Ref attributes.

The attributes for the UALEVEL element are shown in Table 5-6.

**Table 5-8. OMF Attributes for the UALEVEL Element**

Attribute	Brief Description	Format	Description	Req'd?
Ref	Reference to sounding Part	String - "TTAA", "TTBB", etc.	Reference to the part of the sounding from which the level data were derived	Yes
P	Pressure	positive number	Atmospheric pressure at sounding level, in hectoPascals	Yes
H	Geopotential height	Non-negative number of geopotential meters, or 'SURF' for surface, 'TROP' for tropopause, 'MAXW' for level of maximum winds, 'MAXWTOP' for maximum wind level at the top of the sounding, or omitted	Geopotential height of the reported level, or a special height indicator	No
T	Air Temperature	positive, zero, or negative number	Air temperature in degrees Celsius at the reported level	No

**Table 5-8. OMF Attributes for the UALEVEL Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
DP	Dew point temperature	positive, zero, or negative number	Dew point temperature in degrees Celsius at the reported level	No
Wind	Wind speed and direction	String of form "nnn, mm" or "nnn, mm bbb" or "nnn, mm ,aaa" or "nnn, mm bbb, aaa" or omitted	<p>nnn is a true direction <b>from</b> which the wind is blowing, in degrees, or VAR if " the wind is variable, or all directions or unknown or waves confused, direction indeterminate." This is an integer number within [0, 360), with 0 meaning the wind is blowing from true North, 270 stands for the wind blowing from due West. Normally this number has a precision of 10 degrees.</p> <p>mm is the wind speed in meters per second.</p> <p>If specified, bbb stands for the absolute value of the vector difference between the wind at a given level, and the wind 1 km <i>below</i> that level, in meters per second. The number aaa if given is the absolute value of the vector difference between the wind at a given level, and the wind 1 km <i>above</i> that level, in meters per second.</p>	No

**Usage Examples:** See Section 5.3.3.4.4.

#### **OMF DTD for the UALEVEL Element:**

```
<!ELEMENT UALEVEL (#PCDATA)>
<!ATTRIB UALEVEL
  Ref          NMTOKEN      #REQUIRED
  P            NMTOKEN      #REQUIRED
  H            NMTOKEN      #IMPLIED
  T            NMTOKEN      #IMPLIED
  DP           NMTOKEN      #IMPLIED
  Wind         CDATA        #IMPLIED
>
```

##### **5.2.3.5.6 Example of a Complete UAR Report**

```
<UAR TStamp='937850400' LatLon='27.700, -82.400' BId='722100'
SName='KTBW, TAMPA BAY AREA' Elev='12'>
<UAPART id='TTAA'>70181</UAPART>
```

```
<UAPART id='TTAA'>70181</UAPART>
<UAPART id='TTAA'>70181</UAPART>
<UAPART id='TTBB'>70180</UAPART>
<UAPART id='TTBB'>70180</UAPART>
<UAID Ref='TTBB'>45202 81702</UAID>
<UAID Ref='TTBB'>45202 81702</UAID>
<UACODE Ref='TTAA'>51515 10164 00002 10194 14523 16523</UACODE>
<UACODE Ref='TTAA'>51515 10164 00002 10194 14523 16523</UACODE>
<UACODE Ref='TTAA'>51515 10164 00002 10194 14523 16523</UACODE>
<UACODE Ref='TTBB'>41414 885//</UACODE>
<UACODE Ref='TTBB'>41414 885//</UACODE>
<UALEVELS><UALEVEL Ref='TTAA' P='1008.0' H='SURF' T='26.2' DP='4.3' Wind='130, 2'>99008 26243 13003</UALEVEL>
<UALEVEL Ref='TTAA' P='1008.0' H='SURF' T='26.2' DP='4.3' Wind='130, 2'>99008 26243 13003</UALEVEL>
<UALEVEL Ref='TTAA' P='1008.0' H='SURF' T='26.2' DP='4.3' Wind='130, 2'>99008 26243 13003</UALEVEL>
<UALEVEL Ref='TTBB' P='1008.0' H='SURF' T='26.2' DP='4.3'>00008 26243</UALEVEL>
<UALEVEL Ref='TTBB' P='1008.0' H='SURF' T='26.2' DP='4.3'>00008 26243</UALEVEL>
<UALEVEL Ref='TTAA' P='1000.0' H='83' T='25.6' DP='4.1' Wind='130, 4'>00083 25641 13007</UALEVEL>
<UALEVEL Ref='TTAA' P='1000.0' H='83' T='25.6' DP='4.1' Wind='130, 4'>00083 25641 13007</UALEVEL>
<UALEVEL Ref='TTAA' P='1000.0' H='83' T='25.6' DP='4.1' Wind='130, 4'>00083 25641 13007</UALEVEL>
<UALEVEL Ref='TTBB' P='947.0' T='21.4' DP='1.5'>11947 21415</UALEVEL>
<UALEVEL Ref='TTBB' P='947.0' T='21.4' DP='1.5'>11947 21415</UALEVEL>
<UALEVEL Ref='TTBB' P='933.0' T='22.6' DP='5.0'>22933 22650</UALEVEL>
<UALEVEL Ref='TTBB' P='933.0' T='22.6' DP='5.0'>22933 22650</UALEVEL>
<UALEVEL Ref='TTAA' P='925.0' H='765' T='22.2' DP='5.0' Wind='145, 14'>92765 22250 14527</UALEVEL>
<UALEVEL Ref='TTAA' P='925.0' H='765' T='22.2' DP='5.0' Wind='145, 14'>92765 22250 14527</UALEVEL>
<UALEVEL Ref='TTAA' P='925.0' H='765' T='22.2' DP='5.0' Wind='145, 14'>92765 22250 14527</UALEVEL>
<UALEVEL Ref='TTAA' P='850.0' H='1496' T='17.2' DP='2.2' Wind='165, 12'>85496 17222 16525</UALEVEL>
<UALEVEL Ref='TTAA' P='850.0' H='1496' T='17.2' DP='2.2' Wind='165, 12'>85496 17222 16525</UALEVEL>
<UALEVEL Ref='TTAA' P='850.0' H='1496' T='17.2' DP='2.2' Wind='165, 12'>85496 17222 16525</UALEVEL>
<UALEVEL Ref='TTBB' P='820.0' T='15.0' DP='1.0'>33820 15010</UALEVEL>
<UALEVEL Ref='TTBB' P='820.0' T='15.0' DP='1.0'>33820 15010</UALEVEL>
<UALEVEL Ref='TTAA' P='700.0' H='3129' T='7.2' DP='0.8' Wind='175, 13'>70129 07208 17526</UALEVEL>
<UALEVEL Ref='TTAA' P='700.0' H='3129' T='7.2' DP='0.8' Wind='175, 13'>70129 07208 17526</UALEVEL>
<UALEVEL Ref='TTAA' P='700.0' H='3129' T='7.2' DP='0.8' Wind='175, 13'>70129 07208 17526</UALEVEL>
<UALEVEL Ref='TTBB' P='700.0' T='7.2' DP='0.8'>44700 07208</UALEVEL>
<UALEVEL Ref='TTBB' P='700.0' T='7.2' DP='0.8'>44700 07208</UALEVEL>
<UALEVEL Ref='TTBB' P='682.0' T='6.8' DP='3.7'>55682 06837</UALEVEL>
<UALEVEL Ref='TTBB' P='682.0' T='6.8' DP='3.7'>55682 06837</UALEVEL>
<UALEVEL Ref='TTBB' P='657.0' T='6.2' DP='3.7'>66657 06237</UALEVEL>
<UALEVEL Ref='TTBB' P='657.0' T='6.2' DP='3.7'>66657 06237</UALEVEL>
```

```
<UALEVEL Ref='TTBB' P='590.0' T='1.4' DP='3.2'>77590 01432</UALEVEL>
<UALEVEL Ref='TTBB' P='590.0' T='1.4' DP='3.2'>77590 01432</UALEVEL>
<UALEVEL Ref='TTBB' P='553.0' T='-1.4' DP='0.4'>88553 01504</UALEVEL>
<UALEVEL Ref='TTBB' P='553.0' T='-1.4' DP='0.4'>88553 01504</UALEVEL>
<UALEVEL Ref='TTAA' P='500.0' H='5840' T='-4.8' DP='1.0' Wind='240,
10'>50584 04910 24021</UALEVEL>
<UALEVEL Ref='TTAA' P='500.0' H='5840' T='-4.8' DP='1.0' Wind='240,
10'>50584 04910 24021</UALEVEL>
<UALEVEL Ref='TTAA' P='500.0' H='5840' T='-4.8' DP='1.0' Wind='240,
10'>50584 04910 24021</UALEVEL>
<UALEVEL Ref='TTBB' P='470.0' T='-7.2' DP='1.0'>99470 07310</UALEVEL>
<UALEVEL Ref='TTBB' P='470.0' T='-7.2' DP='1.0'>99470 07310</UALEVEL>
<UALEVEL Ref='TTBB' P='415.0' T='-12.2' DP='4.3'>11415 12343</UALEVEL>
<UALEVEL Ref='TTBB' P='415.0' T='-12.2' DP='4.3'>11415 12343</UALEVEL>
<UALEVEL Ref='TTAA' P='400.0' H='7570' T='-14.6' DP='7.0' Wind='355,
8'>40757 14757 35515</UALEVEL>
<UALEVEL Ref='TTAA' P='400.0' H='7570' T='-14.6' DP='7.0' Wind='355,
8'>40757 14757 35515</UALEVEL>
<UALEVEL Ref='TTAA' P='400.0' H='7570' T='-14.6' DP='7.0' Wind='355,
8'>40757 14757 35515</UALEVEL>
<UALEVEL Ref='TTBB' P='383.0' T='-17.0' DP='9.0'>22383 17159</UALEVEL>
<UALEVEL Ref='TTBB' P='383.0' T='-17.0' DP='9.0'>22383 17159</UALEVEL>
<UALEVEL Ref='TTBB' P='347.0' T='-23.6' DP='4.7'>33347 23747</UALEVEL>
<UALEVEL Ref='TTBB' P='347.0' T='-23.6' DP='4.7'>33347 23747</UALEVEL>
<UALEVEL Ref='TTBB' P='312.0' T='-26.8' DP='3.8'>44312 26938</UALEVEL>
<UALEVEL Ref='TTBB' P='312.0' T='-26.8' DP='3.8'>44312 26938</UALEVEL>
<UALEVEL Ref='TTAA' P='300.0' H='9680' T='-28.8' DP='4.0' Wind='240,
8'>30968 28940 24016</UALEVEL>
<UALEVEL Ref='TTAA' P='300.0' H='9680' T='-28.8' DP='4.0' Wind='240,
8'>30968 28940 24016</UALEVEL>
<UALEVEL Ref='TTAA' P='300.0' H='9680' T='-28.8' DP='4.0' Wind='240,
8'>30968 28940 24016</UALEVEL>
<UALEVEL Ref='TTAA' P='250.0' H='10960' T='-38.6' DP='4.5' Wind='230,
20'>25096 38745 23039</UALEVEL>
<UALEVEL Ref='TTAA' P='250.0' H='10960' T='-38.6' DP='4.5' Wind='230,
20'>25096 38745 23039</UALEVEL>
<UALEVEL Ref='TTAA' P='250.0' H='10960' T='-38.6' DP='4.5' Wind='230,
20'>25096 38745 23039</UALEVEL>
<UALEVEL Ref='TTBB' P='250.0' T='-38.6' DP='4.5'>55250 38745</UALEVEL>
<UALEVEL Ref='TTBB' P='250.0' T='-38.6' DP='4.5'>55250 38745</UALEVEL>
<UALEVEL Ref='TTAA' P='200.0' H='12450' T='-51.4' DP='5.0' Wind='230,
22'>20245 51550 23044</UALEVEL>
<UALEVEL Ref='TTAA' P='200.0' H='12450' T='-51.4' DP='5.0' Wind='230,
22'>20245 51550 23044</UALEVEL>
<UALEVEL Ref='TTAA' P='200.0' H='12450' T='-51.4' DP='5.0' Wind='230,
22'>20245 51550 23044</UALEVEL>
<UALEVEL Ref='TTAA' P='150.0' H='14250' T='-68.0' DP='5.0' Wind='255,
20'>15425 68150 25539</UALEVEL>
<UALEVEL Ref='TTAA' P='150.0' H='14250' T='-68.0' DP='5.0' Wind='255,
20'>15425 68150 25539</UALEVEL>
<UALEVEL Ref='TTAA' P='150.0' H='14250' T='-68.0' DP='5.0' Wind='255,
20'>15425 68150 25539</UALEVEL>
<UALEVEL Ref='TTBB' P='141.0' T='-70.8' DP='4.9'>66141 70949</UALEVEL>
<UALEVEL Ref='TTBB' P='141.0' T='-70.8' DP='4.9'>66141 70949</UALEVEL>
<UALEVEL Ref='TTBB' P='127.0' T='-70.2' DP='6.0'>77127 70356</UALEVEL>
<UALEVEL Ref='TTBB' P='127.0' T='-70.2' DP='6.0'>77127 70356</UALEVEL>
<UALEVEL Ref='TTBB' P='105.0' T='-78.6' DP='6.0'>88105 78756</UALEVEL>
```

```
<UALEVEL Ref='TTBB' P='105.0' T='-78.6' DP='6.0'>88105 78756</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>10662 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='TROP' T='-80.4' DP='6.0' Wind='280,
8'>88100 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>88100 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>10662 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='16620' T='-80.4' DP='6.0' Wind='280,
8'>10662 80556 28015</UALEVEL>
<UALEVEL Ref='TTAA' P='100.0' H='TROP' T='-80.4' DP='6.0' Wind='280,
8'>88100 80556 28015</UALEVEL>
<UALEVEL Ref='TTBB' P='100.0' T='-80.4' DP='6.0'>99100 80556</UALEVEL>
<UALEVEL Ref='TTBB' P='100.0' T='-80.4' DP='6.0'>99100 80556</UALEVEL>
</UALEVELS></UAR>
```

### 5.2.3.6 OMF Markup for Bathymetric, Salinity, and Current Reports (BTSC)

The BTSC element defines an observation report on temperature, salinity, and currents at one particular location on the ocean surface, or in subsurface layers. These data are distributed in WMO-306 FM 63 X BATHY, FM 64 IX TESAC, and FM 62 TRACKOB reports. The main component of the BTSC element is a sequence of BTLEVELs that describe physical characteristics of sea water (temperature, salinity, and current) at particular depth levels.

Chapter 4 of NCAR Technical Note 404 covers in great detail the scope of surface and subsurface oceanographic observations, especially in situ observations, which are annotated in this document. "In order to understand oceanic circulation, it is necessary to know the internal distribution of water mass (e.g., density) within the ocean. The distribution of ocean water density is defined by the water temperature, salinity, and pressure or depth. Measurements of these water properties are generally made with vertical profiling instruments lowered from research ships."

Raw BATHY, TESAC, and TRACKOB observation reports are highly encoded, which obscures the meaning of data and impedes their usage. Without consulting special coding tables, the information in these reports is inaccessible. The OMF is to redress this situation by annotating raw reports with decoded information, in conventional and self-evident units.

The OMF markup described in this document only annotates the raw text of oceanographic profile reports, without altering them or omitting a single character. An annotated BTSC document arranges the information in a logical sequence, which is most suitable for further analysis. Yet it is *always* possible to reconstruct the original report in its entirety. In case of a coding or transmission error, the corresponding data are marked with a specific tag. It is up to an application then to try to correct the error and salvage whatever data possible.

The BTSC Element is contained in a <Reports></Reports> container, and in turn serves as a container for BTID, BTCODE, and BTLEVELS elements.

The attributes of the BTSC element are described in Table 5-7.

**Table 5-9. OMF Attributes for the BTSC Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
TStamp	Time Stamp		<----- See Table 5-1 ----->	Yes
LatLon	Latitude and Longitude of observation		<----- See Table 5-1 ----->	Yes
BId	Station identifier group	positive integer	For a buoy or other observation platform, this ID is a combination of a WMO region number, subarea number (per WMO-306 Code Table 0161), and the buoy type and serial number. This information is reported in Section 4 of a BTSC report.  If Section 4 contains a call sign rather than a numerical id, the BId attribute is computed as <code>itoa(1000009 + hc)</code> , where <code>hc</code> is a numerical representation of the call letters considered as a number in radix 36 notation. For example, "0000" hashes to 0, and "zzzz" hashes to 1,679,615. Note this formula makes the BId attribute a unique numeric identifier for the station.	Yes
SName	Call sign	string	Ship's call sign, if reported	Yes
Title	Report type	string	"JJYY" - FM 63 X Ext. BATHY report "KKXX" - FM 64 IX TESAC report "NNXX" - FM 62 TRACKOB report	Yes
Depth	Water depth	positive number	Total water depth at point of observation	No

**Expanded OMF DTD for the BTSC Element:**

```

<!ELEMENT BTSC (BTID, BTCODE?, BTLEVELS)>
<!ATTRIB BTSC
  TStamp      NMOKEN          #REQUIRED
  LatLon     CDATA           #REQUIRED
  BId        NMOKEN          #REQUIRED
  SName      CDATA           #REQUIRED
  Title      (JJXX|KKXX|NNXX) #REQUIRED
  Depth      NMOKEN          #IMPLIED
>

```

### 5.2.3.6.1 The BTID Element

The BTID element annotates identification and position data, which constitute Section 1 of FM 62 - 64 reports. Attributes of this element provide information on instruments and measurement procedures that were used to perform the observation.

Attributes of the BTID element are shown in Table 5-8.

**Table 5-10. OMF Attributes for the BTID Element**

Attribute	Brief Description	Format	Description	Req'd?
DZ	Indicator for digitization	"7" or "8" or omitted	Indicator for method of digitization used in the report ( $k_1$ field). See WMO-306 Code Table 2262. Required for BATHY and TESAC reports	No
Rec	Instrument type code	5-digit code	Code for expendable bathythermograph (XBT) instrument type and fall rate (WMO-306 Code Table 1770)	No
WS	Wind speed units code	"0", "1", "2", "3", or omitted	Indicator for units of wind speed and type of instrumentation ( $i_u$ field). See WMO-306, Code Table 1853.	No
Curr-s	Method of current speed measurement	"2", "3", "4", or omitted	Indicator for the method of current measurement ( $k_5$ field). See WMO-306 Code Table 2266.	No
Curr-d	Indicators for the method of subsurface current measurement	3-digit numerical code	Indicators for the method of subsurface current measurement ( $K_6k_4k_3$ codes). See WMO-306, Code Tables 2267, 2265, and 2264.	No
AV-T	Averaging period for sea temperature	"0", "1", "2", "3", or omitted (if no sea temperature data are reported)	Code for the averaging period for sea temperature ( $m_T$ code). See WMO-306, Code Table 2604	No
AV-SAL	Averaging period for salinity.	"0", "1", "2", "3", or omitted (if no salinity data are reported)	Code for the averaging period for sea salinity ( $m_S$ code). See WMO-306, Code Table 2604	No
AB-Curr	Averaging period for surface current direction and speed	"0", "1", "2", "3", or omitted (if no current data are reported)	Code for the averaging period for surface current direction and speed ( $m_C$ code). See WMO-306, Code Table 2604	No

**Table 5-10. OMF Attributes for the BTID Element**

Attribute	Brief Description	Format	Description	Req'd?
Sal	Method of salinity/depth measurement	"1", "2", "3", or omitted (if no salinity data are reported)	Code for the method of salinity/depth measurement ( $k_2$ code). See WMO-306, Code Table 2263.	No

**Usage Example:**

```
<BTID DZ='8' Rec='05205' WS='0'>20099 0334/ 11848 16654</BTID>
```

**OMF DTD for the BTID Element:**

```
<!ELEMENT BTID (#PCDATA)>
<!ATTLIST BTID
    DZ          (7 | 8)          #IMPLIED
    Rec         NMTOKEN          #IMPLIED
    WS          (0 | 1 | 2 | 3)   #IMPLIED
    Curr-s     (2 | 3 | 4)       #IMPLIED
    Curr-d     NMTOKEN          #IMPLIED
    AV-T       (0 | 1 | 2 | 3)   #IMPLIED
    AV-Sal    (0 | 1 | 2 | 3)   #IMPLIED
    AV-Curr   (0 | 1 | 2 | 3)   #IMPLIED
    Sal        (1 | 2 | 3)       #IMPLIED
>
```

**5.2.3.6.2 The BTITLE Element**

The BTITLE element marks up those tokens of the original message that are unparsed or cannot be parsed. Occasionally some of report's data cannot be parsed because they were wrongly encoded or garbled during transmission.

Regardless of the reason, the unparsed parts of a report are preserved in the BTITLE element and clearly identified so an application may attempt to extract or salvage whatever information possible. The present markup format does not correct or modify these unparsed pieces, and does not annotate them any further. The unparsed tokens identified by a BTITLE element always occur *at the very end* of a report. This implies that there may be at most one BTITLE element within a single BTSC report.

The BTITLE element has no attributes.

**OMF DTD for the BTITLE Element:**

```
<!ELEMENT BTITLE (#PCDATA)>
```

**5.2.3.6.3 The BTLEVELS Element**

The BTLEVELS element is a container for a sequence of BTLEVEL elements, each of which describes conditions at a particular (sub)surface level. The BTLEVELS occur in the

order of increasing depth, starting with D= '0' for the sea surface layer. An optional BTAIR element may be given to tell the temperature and the wind vector of the air above the sea surface.

The BTLEVELS element has no attributes.

### **Usage Example:**

Here is a BTLEVELS element containing both BTAIR and BTLEVEL tags:

```
<BTLEVELS><BTAIR T='27.8' Wind='60, 5.00'>00605 40278</BTAIR>
<BTLEVEL D='2' T='28.9'>02289</BTLEVEL>
<BTLEVEL D='51' T='28.8'>51288</BTLEVEL>
<BTLEVEL D='68' T='27.8'>68278</BTLEVEL>
<BTLEVEL D='94' T='26.5'>94265</BTLEVEL>
<BTLEVEL D='110' T='26.0'>99901 10260</BTLEVEL>
<BTLEVEL D='136' T='23.6'>36236</BTLEVEL>
<BTLEVEL D='143' T='23.3'>43233</BTLEVEL>
<BTLEVEL D='147' T='22.8'>47228</BTLEVEL>
<BTLEVEL D='162' T='22.0'>62220</BTLEVEL>
<BTLEVEL D='184' T='21.1'>84211</BTLEVEL>
<BTLEVEL D='198' T='20.3'>98203</BTLEVEL>
<BTLEVEL D='209' T='19.5'>99902 09195</BTLEVEL>
<BTLEVEL D='213' T='19.2'>13192</BTLEVEL>
<BTLEVEL D='219' T='18.9'>19189</BTLEVEL>
<BTLEVEL D='232' T='18.2'>32182</BTLEVEL>
<BTLEVEL D='235' T='17.8'>35178</BTLEVEL>
<BTLEVEL D='257' T='16.8'>57168</BTLEVEL>
<BTLEVEL D='317' T='15.1'>99903 17151</BTLEVEL>
<BTLEVEL D='335' T='14.1'>35141</BTLEVEL>
<BTLEVEL D='371' T='12.7'>71127</BTLEVEL>
<BTLEVEL D='389' T='12.0'>89120</BTLEVEL>
<BTLEVEL D='430' T='10.5'>99904 30105</BTLEVEL>
<BTLEVEL D='449' T='9.5'>49095</BTLEVEL>
<BTLEVEL D='514' T='7.8'>99905 14078</BTLEVEL>
<BTLEVEL D='547' T='7.0'>47070</BTLEVEL>
<BTLEVEL D='656' T='5.9'>99906 56059</BTLEVEL>
<BTLEVEL D='738' T='5.3'>99907 38053</BTLEVEL>
</BTLEVELS>
```

### **OMF DTD for the BTLEVELS Element:**

```
<!ELEMENT BTLEVELS(BTAIR?, (BTLEVEL)*)>
```

#### **5.2.3.6.4 The BTAIR Element**

The optional BTAIR element describes the air above the sea surface. The body of the element is one or two encoded tokens from Section 1 of FM 62 - 64 reports (the i<sub>u</sub>ddff and 4s<sub>n</sub>TTT groups). The BTAIR element's attributes spell out this information in "plain text".

Table 5-9 describes the attributes of the BTAIR element.

**Table 5-11. OMF Attributes for the BTAIR Element**

<b>Attribute</b>	<b>Brief Description</b>	<b>Format</b>	<b>Description</b>	<b>Req'd?</b>
T	Air temperature	Positive, zero, or negative number, or omitted	Air temperature just above the sea surface, in degrees Celsius.	No
Wind	Wind vector	String of form "nnn,mm", or omitted	Here nnn is a true direction from which the wind is blowing, in degrees, or VAR if " the wind is variable, or all directions or unknown or waves confused, direction indeterminate." This is an integer number within [0, 360), with 0 meaning the wind is blowing from the true North; 270 means the wind is blowing from the West. Normally this number has a precision of 10 degrees.  mm is the wind speed in meters per second.	No

**Usage Example:**

```
<BTAIR T='27.8' Wind='60, 5.00'>00605 40278</BTAIR>
```

**OMF DTD for the BTAIR Element:**

```
<!ELEMENT BTAIR (#PCDATA)>
<!ATTLIST BTAIR
      T          NMTOKEN      #IMPLIED
      Winc       CDATA        #IMPLIED
    >
```

**5.2.3.6.5 The BTLEVEL Element**

The BTLEVEL element conveys detailed information about one vertical level (surface or subsurface) -- both in a raw and decoded formats. The body of the element is encoded information about a depth level exactly as it appeared in a raw report. A required D attribute tells the depth of the level in meters. The other attributes of a BTLEVEL element present the element's encoded data in a fully-decoded, self-contained format.

Table 5-10 describes the attributes of the BTLEVEL element.

**Table 5-12. OMF Attributes for the BTLEVEL Element**

## fnmoc\_metcast\_pg

---

Attribute	Brief Description	Format	Description	Req'd?
D	Depth	Non-negative number	Depth of the level in meters.	Yes
T	Water temperature	Positive, zero, or negative number, or omitted	Water temperature at the reported level.	No
S	Salinity	Positive number, or omitted	Salinity at the reported level, in parts per thousand.	No
C	Current vector	String of form "nnn,mm", or omitted	<p>nnn is the true direction <b>toward</b> which the sea current is moving, in degrees, or VAR if "the current is variable, or all directions or unknown, direction indeterminate." This is an integer number within [ 0 , 360 ), with 0 meaning the current flows toward true North; 270 means the current is flowing toward the West. Normally this number has a precision of 10 degrees.</p> <p>mm is the speed of current in meters per second.</p>	No

### Usage Example:

```
<BTLEVEL D='100' T='18.7'>99901 00187</BTLEVEL>
```

### OMF Markup for the BTLEVEL Element:

```
<!ELEMENT BTLEVEL (#PCDATA)>
<!ATTLIST BTLEVEL
  D          NMTOKEN      #REQUIRED
  T          NMTOKEN      #IMPLIED
  S          NMTOKEN      #IMPLIED
  Curr      CDATA        #IMPLIED
>
```

#### 5.2.3.6.6 Example of a complete BTSC Report

```
<BTSC TStamp='937826280' LatLon='18.283, 163.950' BId='2525997'
SName='WPGK, ' Title='JJYY' Depth='1926'>
<BTID DZ='8' Rec='05205' WS='0'>20099 1118/ 11817 16357</BTID>
<BTLEVELS><BTAIR T='27.8' Wind='60, 5.00'>00605 40278</BTAIR>
<BTLEVEL D='2' T='28.9'>02289</BTLEVEL>
<BTLEVEL D='51' T='28.8'>51288</BTLEVEL>
<BTLEVEL D='68' T='27.8'>68278</BTLEVEL>
<BTLEVEL D='94' T='26.5'>94265</BTLEVEL>
```

```
<BTLEVEL D='110' T='26.0'>99901 10260</BTLEVEL>
<BTLEVEL D='136' T='23.6'>36236</BTLEVEL>
<BTLEVEL D='143' T='23.3'>43233</BTLEVEL>
<BTLEVEL D='147' T='22.8'>47228</BTLEVEL>
<BTLEVEL D='162' T='22.0'>62220</BTLEVEL>
<BTLEVEL D='184' T='21.1'>84211</BTLEVEL>
<BTLEVEL D='198' T='20.3'>98203</BTLEVEL>
<BTLEVEL D='209' T='19.5'>99902 09195</BTLEVEL>
<BTLEVEL D='213' T='19.2'>13192</BTLEVEL>
<BTLEVEL D='219' T='18.9'>19189</BTLEVEL>
<BTLEVEL D='232' T='18.2'>32182</BTLEVEL>
<BTLEVEL D='235' T='17.8'>35178</BTLEVEL>
<BTLEVEL D='257' T='16.8'>57168</BTLEVEL>
<BTLEVEL D='317' T='15.1'>99903 17151</BTLEVEL>
<BTLEVEL D='335' T='14.1'>35141</BTLEVEL>
<BTLEVEL D='371' T='12.7'>71127</BTLEVEL>
<BTLEVEL D='389' T='12.0'>89120</BTLEVEL>
<BTLEVEL D='430' T='10.5'>99904 30105</BTLEVEL>
<BTLEVEL D='449' T='9.5'>49095</BTLEVEL>
<BTLEVEL D='514' T='7.8'>99905 14078</BTLEVEL>
<BTLEVEL D='547' T='7.0'>47070</BTLEVEL>
<BTLEVEL D='656' T='5.9'>99906 56059</BTLEVEL>
<BTLEVEL D='738' T='5.3'>99907 38053</BTLEVEL>
</BTLEVELS></BTSC>
```

### 5.2.3.7 OMF Markup for a Terminal Aerodrome Forecast (TAF)

A TAF forecast is contained within a TAF element, which serves as a container for VALID and PERIOD elements. The TAF element itself is wrapped in a <FORECASTS></FORECASTS> container.

The TAF element defines one particular TAF forecast. This element contains (in its child elements) the raw text of a forecast, a sequence of words as they appear in a TAF bulletin. Groups of related words are annotated in the element attributes to tell their type and to communicate their meaning in a more uniform and machine-friendly way. For example, time stamps are always presented in EPOCH seconds, airport references are annotated with the airport's full name and location, and forecast change periods are clearly separated and timestamped.

The body of the TAF element is empty. The element's attributes are shown in Table 5-11.

**Table 5-13. OMF Attributes for the TAF Element**

Attribute	Brief Description	Format	Description	Req'd?
TStamp	Time Stamp	<----- See Table 5-1 ----->		Yes
LatLon	Latitude and Longitude of observation	<----- See Table 5-1 ----->		Yes

**Table 5-13. OMF Attributes for the TAF Element**

Attribute	Brief Description	Format	Description	Req'd?
BId	Block Station ID	positive integer	WMO Block Station ID of the reporting station	Yes
SName	Call sign	string	Ship's call sign, if reported	Yes

**Usage Example:**

```
<TAF TStamp='937935300' LatLon='24.550, -81.750' BId='722010'  
SName='KEYW, KEY WEST INTL ARPT'>  
.  
.  
.</TAF>
```

**OMF Markup for the TAF Element:**

```
<!ELEMENT TAF (VALID, PERIOD+)>  
<!ATTLIST TAF  
    TStamp      NMOKEN      #REQUIRED  
    LatLon     CDATA       #REQUIRED  
    BId        NMOKEN      #REQUIRED  
    SName      CDATA       #REQUIRED  
>
```

**5.2.3.7.1 The VALID Element**

The VALID element specifies the period for which the forecast is valid. The body contains the YYGGggZ (report time) and Y<sub>1</sub>Y<sub>1</sub>G<sub>1</sub>G<sub>1</sub>G<sub>2</sub>G<sub>2</sub> (beginning and ending valid times) groups from the TAF. The only attribute is a TRange that describes the beginning and ending valid times of the forecasts in epoch times (seconds since 00:00Z 1 January 1970).

**Usage Example:**

```
<VALID TRange='937936800, 938023200'>211735Z 211818</VALID>
```

This is for a report whose beginning valid time is 1800Z 21 September 1999, and whose ending valid time is 1800Z 22 September 1999. The TRange attribute provides these times in epoch seconds.

**Expanded OMF Markup for the VALID Element:**

```
<!ELEMENT VALID (#PCDATA)>  
<!ATTLIST VALID  
    TRange      CDATA       #REQUIRED  
>
```

### **5.2.3.7.2 The PERIOD Element**

The PERIOD element specifies the complete set of weather condition elements that are forecast for a specified period. The set includes surface wind, visibility, significant weather, clouds and obscurations, and, when expected, non-convective low-level wind shear.

A TAF element contains at least one PERIOD element for the initial period of the forecast. If prevailing conditions are expected to change significantly, additional PERIOD elements may follow.

The PERIOD element is a container for a PREVAILING element and one or more VAR elements. It has two attributes: a mandatory TRange specifying the time period for which the conditions are forecast and an optional Title, which is a forecast change indicator that introduces the present forecast period. The Title, for example, might be a FM group of the TAF report that begins a new subdivision of the forecast, when a significant change in the weather is expected. The Title attribute is always omitted for the initial forecast period.

#### **Usage Examples:**

```
<PERIOD TRange='937936800, 937951200'>
<PERIOD TRange='937951200, 937965600' Title='FM2200'>
```

#### **OMF Markup for the PERIOD Element:**

```
<!ELEMENT PERIOD(PREVAILING, VAR*)>
<!ATTLIST PERIOD
      TRange          NMOKEN      #REQUIRED
      Title           NMOKEN      #IMPLIED
>
```

### **5.2.3.7.3 The PREVAILING Element**

The complete set of forecast weather condition elements for the given period, as it appears in the text of a forecast. This element contains undecoded text from the forecast message. The PREVAILING element is contained within a PERIOD element.

#### **Usage Example:**

```
<PREVAILING>26018G25KT P6SM SCT015 BKN080</PREVAILING>
```

#### **OMF DTD for the PREVAILING Element:**

```
<!ELEMENT PREVAILING (#PCDATA)>
```

### **5.2.3.7.4 The VAR Element**

The VAR element describes variations or fluctuations in forecast meteorological conditions that are expected to occur or develop within a forecast period. This forecast period is specified by the parent (containing) PERIOD element. There may be multiple VAR elements within a single PERIOD element. The content of the VAR element is weather conditions groups (tokens) as they appear in the raw forecast text. This set of weather condition elements is not complete: only the weather element tokens that are expected to change or fluctuate are specified. The VAR element has two attributes: a TRange specifying the time period during which the variations are forecast to occur, and a Title, which is a forecast change indicator that introduces the present variation in the TAF report. The title might consist of "BECMG" (becoming), "TEMPO" (temporarily), or "PROBxx" (probability xx per cent), followed by two 2-digit numbers denoting the beginning and ending hours for the forecast variations.

#### **Usage Examples:**

```
<PERIOD TRange='937951200, 937965600' Title='FM2200'>
<PREVAILING>26018G25KT P6SM SCT015 BKN080</PREVAILING>
<VAR Title='TEMPO 2202' TRange='937951260, 937965600'>3SM SHRA
BKN015</VAR>
</PERIOD>
<PERIOD TRange='937965600, 938001600' Title='FM0200'>
<PREVAILING>29015KT P6SM SCT015</PREVAILING>
<VAR Title='PROB40 0206' TRange='937965660, 937980000'>3SM SHRA
BKN015</VAR>
</PERIOD>
```

Here the first period runs from 2200 to 0200. The forecast for this period is for winds from the WSW (260) at 18 knots with gusts to 25, visibility greater than 6 statute miles, with a scattered cloud deck at 1500 feet and broken clouds at 8000 feet. The VAR tag for this period runs from 2200 to 0200 and forecasts temporary periods of reduced visibility (3 statute miles) in showers and rain, with a broken cloud deck at 1500 feet.

For the second period (0200 to 1200), the forecast is for winds from the WNW (290) at 15 knots, visibility greater than 6 statute miles, and scattered clouds at 1500 feet. The VAR indicator for this period indicates that from 0200 to 0600 there will be a 40 per cent probability of reduced visibility (3 statute miles) and broken cloudiness at 1500 feet during showers and rain.

#### **OMF DTD for the VAR Element:**

```
<!ELEMENT VAR (#PCDATA)>
<!ATTLIST VAR
    TRange      NMOKEN      #REQUIRED
    Title       CDATA        #REQUIRED
>
```

### 5.2.3.7.5 Example of a Complete TAF Report

```
<TAF TStamp='937935300' LatLon='24.550, -81.750' BId='722010'
SName='KEYW, KEY WEST INTL ARPT'>
<VALID TRange='937936800, 938023200'>211735Z 211818</VALID>
<PERIOD TRange='937936800, 937951200'>
<PREVAILING>24020G30KT P6SM SCT015 BKN050 OVC150</PREVAILING>
<VAR Title='TEMPO 1822' TRange='937936860, 937951200'>2SM SHRA
BKN015</VAR>
</PERIOD>
<PERIOD TRange='937951200, 937965600' Title='FM2200'>
<PREVAILING>26018G25KT P6SM SCT015 BKN080</PREVAILING>
<VAR Title='TEMPO 2202' TRange='937951260, 937965600'>3SM SHRA
BKN015</VAR>
</PERIOD>
<PERIOD TRange='937965600, 938001600' Title='FM0200'>
<PREVAILING>29015KT P6SM SCT015</PREVAILING>
<VAR Title='PROB40 0206' TRange='937965660, 937980000'>3SM SHRA
BKN015</VAR>
</PERIOD>
<PERIOD TRange='938001600, 938023200' Title='FM1200'>
<PREVAILING>31012KT P6SM SCT025</PREVAILING>
</PERIOD></TAF>
```

### 5.2.3.8 OMF Markup for SIGMETs

The SIGMET element defines one particular SIGMET advisory. The element's content is a raw report: a sequence of lines as they appear in a bulletin. The lines are annotated to tell their type and to communicate their meaning in a more uniform and machine-friendly way: For example, time stamps are always presented in EPOCH seconds, airport references are converted into points on the globe specified as lat,lon pairs.

SIGMET elements are wrapped in an <Advisories></Advisories> container, and in turn serve as a containers for VALID, AFFECTING, EXTENT, and BODY elements.

Table 5-12 describes the attributes for the SIGMET element.

**Table 5-14. OMF Attributes for the SIGMET Element**

Attribute	Brief Description	Format	Description	Req'd?
class	SIGMET type	"CONVECTIVE", "HOTEL", "INDIA", "UNIFORM", "VICTOR", "WHISKEY"	Identifier for the type of SIGMET message	Yes
id	Identifier for a particular advisory	String	Identifier for the advisory; value depends on the advisory class.	Yes

**Table 5-14. OMF Attributes for the SIGMET Element**

Attribute	Brief Description	Format	Description	Req'd?
TStamp	Time Stamp	<----- See Table 5-1 ----->		Yes
BBox	Bounding box for advisory area	<----- See Table 5-1 ----->		Yes

**Usage Example:**

```
<SIGMET class='CONVECTIVE' id='61E' TStamp='937857300'>
.
.
.
</SIGMET>
```

**OMF Markup for the SIGMET Element:**

```
<!ELEMENT SIGMET(VALID, AFFECTING?, EXTENT, BODY)>
<!ATTLIST SIGMET
    class (CONVECTIVE|HOTEL|INDIA|UNIFORM|VICTOR|WHISKEY) #REQUIRED
    id          NMTOKEN #REQUIRED
    TStamp      NMTOKEN #REQUIRED
    BBox        CDATA #REQUIRED
>
```

**5.2.3.8.1 The VALID Element**

The VALID element specifies the period for which the advisory is valid. The body contains the valid time specification from the advisory. The only attribute is a TRange that describes the beginning and ending valid times of the forecasts in epoch times (seconds since 00:00Z 1 January 1970).

**Usage Example:**

```
<VALID TRange='937857300, 937864500'>VALID UNTIL 2155Z</VALID>
```

**Expanded OMF Markup for the VALID Element:**

```
<!ELEMENT VALID (#PCDATA)>
<!ATTLIST VALID
    TRange      CDATA #REQUIRED
>
```

**5.2.3.8.2 The AFFECTING Element**

The AFFECTING element contains a brief description of the area affected by the advisory. The body contains the area description from the advisory. There are no attributes for this element.

**Usage Example:**

```
<AFFECTING>GA FL AND SC GA FL CSTL WTRS</AFFECTING>
```

**OMF DTD for the AFFECTING Element:**

```
<!ELEMENT AFFECTING (#PCDATA)>
```

**5.2.3.8.3 The EXTENT Element**

The EXTENT element gives a precise specification of the area affected by the condition for which the present advisory was issued. The body of the element contains the corresponding FROM record of a raw report. The attributes normalize the coordinates in a lat,lon format.

The attributes for the EXTENT element are described in Table 5-13.

**Table 5-15. OMF Attributes for the EXTENT Element**

Attribute	Brief Description	Format	Description	Req'd?
Shape	Type of area specification	"AREA", "LINE", "POINT"	Type of area shape specified	Yes
LatLons	List of latitudes and longitudes defining the area	Positive, zero, or negative numbers in lat/lon pairs	Control points (vertices) for a polygon/polyline representing the affected area	Yes

**Usage Examples:**

```
<EXTENT Shape='AREA' LatLons='28.706, -85.163, 28.217, -81.430, 25.211, -80.934, 24.104, -82.955, 28.706, -85.163'>110SSW TLH-20S ORL-50SW MIA-70WSW EYW-110SSW TLH</EXTENT>
```

```
<EXTENT Shape='LINE' LatLons='39.121, -78.384, 37.780, -80.487'>40NW CSN-30E BKW</EXTENT>
```

**OMF DTD for the EXTENT Element:**

```
<!ELEMENT EXTENT (#PCDATA)>
<!ATTLIST EXTENT
      Shape          (AREA|LINE|POINT)      #REQUIRED
      LatLons        CDATA                 #REQUIRED
    >
```

**5.2.3.8.4 The BODY Element**

The BODY element contains the remainder of the advisory, exactly as it appears in a bulletin. This element has no attributes.

**Usage Examples:**

```
<BODY>AREA EMBD TS MOV FROM 25020KT. TOPS TO FL350.  
</BODY>  
  
<BODY>DVLPG LINE TS 10 NM WIDE MOV FROM 24025KT. TOPS TO FL330.  
</BODY>
```

**OMF DTD for the BODY Element:**

```
<!ELEMENT BODY (#PCDATA)>
```

**5.2.3.8.5 Example of a Complete SIGMET Report**

```
<SIGMET class='CONVECTIVE' id='61E' TStamp='937857300'>  
<VALID TRange='937857300, 937864500'>VALID UNTIL 2155Z</VALID>  
<AFFECTING>GA FL AND SC GA FL CSTL WTRS</AFFECTING>  
<EXTENT Shape='AREA' LatLons='32.900, -78.640, 28.224, -78.856, 28.112,  
-80.204, 30.754, -83.940, 32.900, -78.640'>70E CHS-90ENE VRB-30NNE VRB-  
30NE TLH-70E CHS</EXTENT>  
<BODY>AREA EMBD TS MOV FROM 19015KT. TOPS TO FL400.  
</BODY>  
</SIGMET>
```

**5.2.3.9 OMF Markup for an AIRMET Report**

The AIRMET element defines one particular AIRMET advisory. The element content is a raw report: a sequence of lines as they appear in a bulletin. The lines are annotated to tell their type and to communicate their meaning in a more uniform and machine-friendly way: For example, time stamps are always presented in EPOCH seconds, airport references are converted into points on the globe specified as lat, lon pairs.

AIRMET elements are wrapped in an <Advisories></Advisories> container, and in turn serve as a containers for TYPE, AFFECTING, EXTENT, and BODY elements.

Table 5-16 describes the attributes for the AIRMET element.

**Table 5-16. OMF Attributes for the AIRMET Element**

Attribute	Brief Description	Format	Description	Req'd?
class	AIRMET type	"SIERRA", "TANGO", "ZULU"	Identifier for the type of AIRMET message: SIERRA = IFR/Mountain Obscuration, TANGO = Turbulence, ZULU = Icing	Yes
TStamp	Time Stamp	<----- See Table 5-1 ----->		Yes
SName	Call sign (ICAO id) and full name of a station	<----- See Table 5-1 ----->		Yes

**Table 5-16. OMF Attributes for the AIRMET Element**

Attribute	Brief Description	Format	Description	Req'd?
Bid	Station identification number	Unsigned integer	WMO Block Station ID, or other identifier	Yes
WMO Header	WMO header of bulletin containing the AIRMET	Example: BOSZ WA 091345		Yes

**Usage Example:**

```
<AIRMET Class='SIERRA' SName='KBOS, BOSTON/LOGAN INTL &' Bid='725090'
          TStamp='937857300' WMOHeader='BOSS WA 091345'>
.
.
.
</AIRMET>
```

**OMF DTD for the AIRMET Element:**

```
<!ELEMENT AIRMET (HEADER?, VALID, AIPART+)>
<!ATTLIST AIRMET
      class (SIERRA|TANGO|ZULU)                      #REQUIRED
      %TStamp           NMTOKEN                         #REQUIRED
      %SName            NMTOKEN                         #REQUIRED
      %Bid              NMTOKEN                         #REQUIRED
      %WMOHeader;       NMTOKEN                         #REQUIRED
>
```

**5.2.3.9.1 The HEADER Element**

The HEADER element annotates the AIRMET data that indicates the type of the information presented in it. This element has a single attribute: Update. If the AIRMET advisory is not an update to a previous advisory, then this attribute is omitted.

**Usage Example:**

```
<HEADER Update='2'>AIRMET UPDT 2 FOR IFR AND MT OBSCN</HEADER>
```

**OMF DTD for the AIRMET Element:**

```
<!ELEMENT HEADER (#PCDATA)
<!ATTLIST HEADER
      Update        NMTOKEN      #IMPLIED
>
```

### **5.2.3.9.2 The VALID Element**

The VALID element specifies the time period for which the advisory is valid. The body contains the valid time range that is specified in the advisory. The only attribute is a TRange, which describes the beginning and ending times of the forecast period in epoch time (seconds since 00:00Z 01 January 1970)

#### **Usage Example:**

```
<VALID TRange='894934500, 894935560'>VALID UNTIL 092000</VALID>
```

#### **OMF DTD for the VALID Element:**

```
<!ELEMENT VALID (#PCDATA)
<!ATTLIST VALID
      TRange      CDATA          #REQUIRED
>
```

### **5.2.3.9.3 The AIPART Element**

The AIPART element presents a single part of an AIRMET. There may be one or more AIPART elements in a given AIRMET. Each part annotated by this element contains AIRMET data collected for a specific purpose for a given geographic area. This information is annotated with the help of four elements internal to the AIPART element. These elements are TYPE, AFFECTING, EXTENT, and BODY.

#### **Usage Example for the AIPART Element:**

See the “Example of a Complete AIRMET Report” listed on the next page for an example of this Element.

#### **OMF DTD for the AIPART Element:**

```
<!ELEMENT AIPART (TYPE?, AFFECTING?, EXTENT? BODY) >
```

#### **The TYPE Element**

The TYPE element annotates the type of the phenomenon described by the AIRMET segment. Possible values include IFR and MTN OBSCN for AIRMET Sierra, TURB for AIRMET Tango, and ICE and FRZLVL for AIRMET Zulu. The element is optional and is omitted if no type information is present in the AIRMET segment.

#### **Usage Example for the TYPE Element:**

```
<TYPE>AIRMET MTN OBSCN</TYPE>
```

#### **OMF DTD for the TYPE Element:**

```
<!ELEMENT TYPE (#PCDATA) >
```

## **The AFFECTING Element**

The AFFECTING element presents a brief description of the affected area

### **Usage Example:**

```
<AFFECTING>ME NH VT NY AND CSTL WTRS</AFFECTING>
```

### **OMF DTD for the AFFECTING Element:**

```
<!ELEMENT AFFECTING (#PCDATA) >
```

## **The EXTENT Element**

The EXTENT element gives a precise specification of the area affected by a weather condition for which the present advisory was issued. The content of the element is the corresponding record of a raw report. A single required attribute, **LatLon**, normalizes the coordinates into latitude, longitude format using control points (vertices) for a polygon/polyline representing the affected area.

### **Usage Example:**

```
<EXTENT LatLons='28.706, -85.163, 28.217, -81.430, 25.211, -80.934,  
24.104, -82.955, 28.706, -85.163'>110SSW TLH-20S ORL-50SW MIA-70WSW  
EYW-110SSW TLH</EXTENT>
```

### **OMF DTD for the EXTENT Element:**

```
<!ELEMENT EXTENT (#PCDATA)>  
<!ATTLIST EXTENT  
      LatLon      CDATA      #REQUIRED  
>
```

## **The BODY Element**

The BODY element annotates the remainder of the advisory, exactly as it appears in a bulletin. This element has no attributes.

### **Usage Examples:**

```
<BODY>OCNL MOD RIME MXD ICGICIP BTN 100 AND FL200. CONDS CONTG BYD 20Z  
THRU 02Z.  
</BODY>
```

### **OMF DTD for the BODY Element:**

```
<!ELEMENT BODY (#PCDATA)>
```

#### **5.2.3.9.4 Example of a Complete AIRMET Report**

```
<Advisories TStamp="894934500">  
  
<AIRMET Class='SIERRA' SName='KBOS, BOSTON/LOGAN INTL &' BId='725090'
```

```
TStamp='894934500' WMOHeader='BOSS WA 091345'>
<HEADER Update='2'>AIRMET UPDT 2 FOR IFR AND MTN OBSCN</HEADER>
<VALID TRange='894934500, 894935560'>VALID UNTIL 092000</VALID>
<AIPART>
  <TYPE>AIRMET IFR</TYPE>
  <AFFECTING>ME NH VT NY AND CSTL WTRS</AFFECTING>
  <EXTENT LatLons='41.509, -88.534, 38.030, -88.588, 35.360, -91.325,
    35.397, -92.230, 40.191, -90.352, 41.509, -88.534'>FROM 50N PQI TO
    YSJ TO 120SE BGR TO 15S CON TO MPV TO MSS TO YQB TO 50N PQI
  </EXTENT>
  <BODY>
    OCNL CIG BLW 010/VIS BLW 3SM PCPN BR. CONDS NY VT AND NH PTNS 16Z-
      18Z. CONTG ME AND CSTL WTRS BYD 20Z THRU 02Z.
  </BODY>
</AIPART>
<AIPART>
  <TYPE>AIRMET MTN OBSCN</TYPE>
  <AFFECTING>ME NH VT NY</AFFECTING>
  <EXTENT LatLons='42.509, -89.534, 39.030, -89.588, 38.360, -90.325,
    30.397, -91.230, 32.191, -88.352'>
    FROM PQI TO MLT TO CON TO ALB TO SYR TO YOW TO PQI
  </EXTENT>
  <BODY>
    MTNS OCNL OBSC IN CLDS/PCPN. CONDS CONTG BYD 20Z THRU 02Z.
  </BODY>
</AIPART>
</AIRMET>
</ADVISORIES>
```

### **5.2.3.10 OMF Markup for Plain Text Messages**

The `Messages` element defines a collection of plain-text WMO Meteorological messages: tropical weather discussions, marine weather summaries, notices to field, recreational forecasts, etc. It may also include messages that are collectives of decodable reports such as METARs, TAFs, etc. The present markup encapsulates the body of a message as it is, preserving line boundaries and empty lines. The markup only adds annotations regarding the message type, an originating center and the time of issue.

The `Messages` element has only one attribute, `TStamp`, which is the time at which the collection of messages was created, in epoch time (seconds since 0000Z 1 January 1970). It serves as a container for `<MSG>` elements containing the individual messages.

#### **Usage Example:**

```
<Messages TStamp="939857994">
<MSG ...>...</MSG>
<MSG ...>...</MSG>
.
.
.
</Messages>
```

#### **OMF DTD for the Messages Element:**

```
<!ELEMENT Messages MSG* >
<!ATTLIST Messages %TStamp;>
```

### 5.2.3.10.1 The **MSG** Element

A MSG element carries one meteorological message or notice. The element's content -- a CDATA section -- is a raw message: a sequence of lines as they appear in a bulletin. All line breaks and empty lines are fully preserved. The message is annotated with meta-information, which defines message's type and the circumstances of its issue. The attributes of the MSG element are described in Table 5-17.

**Table 5-17. OMF Attributes for the MSG Element**

Attribute	Brief Description	Format	Description	Req'd?
id	Message identifier	A NMOKEN, a four-to-six-character string of a form T <sub>1</sub> T <sub>2</sub> A <sub>1</sub> A <sub>2</sub> ii	Designator for the message type and subtype (T <sub>1</sub> T <sub>2</sub> ), area (A <sub>1</sub> A <sub>2</sub> ), and sequence code (ii) of the message, as described in WMO-386.	Yes
Type	Message type	2-letter string (T <sub>1</sub> T <sub>2</sub> )	Designator for the message type and subtype (T <sub>1</sub> T <sub>2</sub> ) as specified in WMO-386, Tables A and B1 through B6	Yes
TStamp	Time Stamp	<----- See Table 5-1 ----->		Yes
SName	Originating station name	String	String containing the identification of the station that originated the message (normally its ICAO call sign)	Yes
BBB	Annotation group	3-character string	So-called "BBB groups" from the abbreviated message line. They indicate that the message has been delayed, corrected or amended. A BBB group can also be used for segmentation. See the WMO-386 for more detail.	No
Descr	Description	String	Keywords and other information describing the message.	No
BBox	Bounding box	<----- See Table 5-1 ----->		No

### Usage Example:

```
<MSG id="WWUE5" Type='WW' TStamp="939826740" SName='KAWN, '
Descr='Weather Warnings'>
<![CDATA[WWUS 35 KJAX 131459
SPSJAX
AMZ435-450-452-454-470-472-474-FLZ020>025-030>033-035>038-040-
GAZ132>136-149>154-162>166-132200-
```

NORTHEAST FLORIDA AND SOUTHEAST GEORGIA HAZARDOUS WEATHER OUTLOOK  
NATIONAL WEATHER SERVICE JACKSONVILLE FL  
1100 AM EDT WED OCT 13 1999

NO HAZARDOUS WEATHER IS EXPECTED ACROSS SOUTHEAST GEORGIA OR  
NORTHEAST FLORIDA THIS AFTERNOON. THEREFORE SPOTTER ACTIVATION IS NOT  
ANTICIPATED.

FOR ADDITIONAL INFORMATION...VISIT THE NATIONAL WEATHER SERVICE  
JACKSONVILLE WEBSITE ON THE INTERNET AT [HTTP://WWW.NWSJAX.NOAA.GOV](http://www.nwsjax.noaa.gov)  
(ALL IN LOWER CASE).

SANTOS  
]]></MSG>

**OMF DTD for the MSG Element:**

```
<!ELEMENT MSG ANY >
<!ATTLIST MSG
  id      NMOKEN #REQUIRED
  Type    NMOKEN #IMPLIED
  %TStamp; %SName; %BBox-OPT;
  BBB     CDATA   #IMPLIED
  Descr   CDATA   #IMPLIED
>
```

### **5.2.3.10.2 Example of a Complete Messages Collection**

```
<!DOCTYPE OMF SYSTEM "http://zowie.metnet.navy.mil/~spawar/JMV-
TNG/XML/OMF.dtd">
<Messages TStamp="939857994">

<MSG id="FAUS5" Type='FA' TStamp="939775500" SName='KBOS, '
BBox='48, -85, 35, -65' Descr='12-HR Aviation Area Forecast and 6-HR
Outlook'>
<![CDATA[FA1W
BOSC FA 130045
SYNOPSIS AND VFR CLDS/WX
SYNOPSIS VALID UNTIL 131900
CLDS/WX VALID UNTIL 131300...OTLK VALID 131300-131900
ME NH VT MA RI CT NY LO NJ PA OH LE WV MD DC DE VA AND CSTL WTRS
.
SEE AIRMET SIERRA FOR IFR CONDS AND MTN OBSCN.
TS IMPLY SEV OR GTR TURB SEV ICE LLWS AND IFR CONDS.
NON MSL HGTS DENOTED BY AGL OR CIG.
.
SYNOPSIS...01Z HIGH PRES OVR THE NRN/MIDDLE ATLC CST STATES. BY
19Z...CDFNT EXTDG NE-SW ACRS EXTRM NWRN OH. WRMFNT ALG A YSC-ENE-
ACK LN. ...REYNOLDS...
]]></MSG>

<MSG id="UAXX1" Type='UA' TStamp="939772800" SName='KAWN, '
Descr='PIREP' BBB='RTD06'>
<![CDATA[BLI UA /OV BLI150005/TM 0035/FL040/TP BE58/TB CONT MOD 040-SFC=
090; RIME
090=
]]></MSG>

<MSG id="WWUE5" Type='WW' TStamp="939826740" SName='KAWN, '
Descr='Weather Warnings'>
<![CDATA[WWUS 35 KJAX 131459
SPSJAX
AMZ435-450-452-454-470-472-474-FLZ020>025-030>033-035>038-040-
GAZ132>136-149>154-162>166-132200

NORTHEAST FLORIDA AND SOUTHEAST GEORGIA HAZARDOUS WEATHER OUTLOOK
NATIONAL WEATHER SERVICE JACKSONVILLE FL
1100 AM EDT WED OCT 13 1999

NO HAZARDOUS WEATHER IS EXPECTED ACROSS SOUTHEAST GEORGIA OR
NORTHEAST FLORIDA THIS AFTERNOON. THEREFORE SPOTTER ACTIVATION IS NOT
ANTICIPATED.

FOR ADDITIONAL INFORMATION...VISIT THE NATIONAL WEATHER SERVICE
JACKSONVILLE WEBSITE ON THE INTERNET AT HTTP://WWW.NWSJAX.NOAA.GOV
(ALL IN LOWER CASE).

SANTOS
]]></MSG>

</Messages>
```

This page intentionally left blank.

## **Appendix A            PARAMETER SPECIFICATIONS**

The tables below provide usage guidelines, acceptable values, and examples for use of the product specification parameters. The values given represent those in use at the time of publication, and are subject to change. The Dynamic Product List, contained in the METCAST Client's <servername>.xml file, provides a listing of all items currently in the database for a given server.

Note that all values (except level numbers) may be specified in either integer or floating point form (e.g. 500 or 500.0).

### **A.1      block\_id**

The **block\_id** parameter specifies the WMO Block Station Number of a reporting station.

Example: (block\_id 724915) for Monterey, CA.

### **A.2      bounding-box**

The **bounding-box** parameter specifies the latitudinal and longitudinal boundaries of an area of interest. The format is (bounding-box *N-LAT W-LON S-LAT E-LON*), where N-LAT is the northernmost latitude of the area, W-LON the westernmost longitude, S-LAT the southernmost latitude, and E-LON the easternmost longitude. The values may be expressed as numbers, with north latitude being positive, south latitude negative, east longitude positive, and west longitude negative, or as numbers followed by an identifying letter for the hemisphere (e.g 70.0N).

Examples: (bounding-box 60N 140W 20S 100W)

(bounding-box 60.0 -140.0 -20.0 -100.0)

The two examples above are exactly equivalent. Note that the numbers may be expressed either as integers or as floating-point values.

### **A.3      call\_id**

The **call\_id** parameter specifies the ICAO call sign for a land station, call sign or an alphanumerical id for a ship, or a serial number for a buoy. All letters must be in upper case.

Examples: (call\_id "KMRY")

(call\_id " WST9756" )

## A.4 center-id and subcenter-id

The following table contains the list of known production centers and subcenters as described in the GRIB Specification Table 0. The Navy has not yet specified sub-center identifiers. When this occurs, updates will be made to this table. Note that there are typically several subcenters associated with each center ID.

**Table A-1. center-id and subcenter-id codes**

Center ID	Subcenter ID	Production Center Description
7	0	US Weather Service - National Met. Center
7	1	US National Center for NCEP Re-Analysis Project
7	2	US National Center for NCEP Ensemble Products
7	3	US National Center for NCEP Central Operations
7	4	US National Environmental Modeling Center
7	5	US National Hydrometeorological Prediction Center
7	6	US National Marine Prediction Center
7	7	US National Climate Prediction Center
7	8	US National Aviation Weather Center
7	9	US National Storm Prediction Center
7	10	US National Tropical Predication Center
8	0	US Weather Service - NWS Telecomms Gateway
9	0	US Weather Service - Field Stations
9	150	ABRFC - Arkansas-Red River RFC, Tulsa, OK
9	151	Alaska RFC Anchorage, AK
9	152	CBRFC - Colorado Basin RFC, Salt Lake City, UT
9	153	CNRFC - California-Nevada RFC, Sacramento, CA
9	154	LMRFC - Lower Mississippi RFC, Slidel, LA
9	155	MARFC - Middle Atlantic RFC, State College, PA
9	156	MBRFC - Missouri Basin RFC, Kansas City, MO
9	157	NCRFC - North Central RFC, Minneapolis, NM
9	158	NERFC - Northeast RFC, Hartford, CT
9	159	NWRFC - Northwest RFC, Portland, OR
9	160	OHRFC - Ohio Basin RFC, Cincinnati, OH
9	161	SERFC - Southeast RFC, Atlanta, GA

**Table A-1. center-id and subcenter-id codes**

<b>Center ID</b>	<b>Subcenter ID</b>	<b>Production Center Description</b>
9	162	WGRFC - West Gulf RFC, Fort Worth, TX
9	170	OUN - Norman OK WFO
34	0	Japanese Meteorological Agency - Tokyo
52	0	National Hurricane Center, Miami
54	0	Canadian Meteorological Service - Montreal
57	0	US Air Force - Global Weather Center
57	10	US Air Force - TBD
58	0	US Navy - Fleet Numerical Oceanography Center
59	0	NOAA Forecast Systems Lab, Boulder CO
60	0	National Center for Atmospheric Research
74	0	UK Met Office - Bracknell
85	0	French Weather Service - Toulouse
97	0	European Space Agency
98	0	European Center for Medium-Range Weather Forecasts - Reading
99	0	DeBilt, Netherlands

## A.5 depth-max

The `depth-max` parameter specifies the depth, in meters, for the bottommost vertical layer to be returned by a query (e.g., a BTSC product query). The format is `(depth-max depth-max-limit)`, where *depth-max-limit* is the limiting depth.

Example: `(depth-max 1500)`

## A.6 depth-min

The `depth-min` parameter specifies the depth, in meters, for the topmost vertical layer to be returned by a query (e.g., a BTSC product query). The format is `(depth-min depth-min-limit)`, where *depth-min-limit* is the limiting depth.

Example: `(depth-min 100)`

## A.7 product-id

**Table A-2. product-id codes**

product-id	Description
BTSC	Bathythermograph, salinity, conductivity observations
grib	Gridded data in WMO Gridded Binary (GRIB) format, followed by product specifications, (e.g. (grib (product-GRIB-code 7) (layer isobar 500) (tau 0 12 24 36 48 72) (source 58) (process-id 22) (resolution 0.200))
imagery	Satellite imagery (followed by a productname S-expression that specifies area_channel_originator.imagetype (e.g. (imagery (productname "Epac-COAM27k_H85Ghz_FNMOC.jif"))
METAR	Hourly/special surface meteorological reports
PIREP	Pilot Weather Report
SIGMET	Significant Meteorological Bulletins
AIRMET	AIRMAN's Meteorological Notice
SYNSEA	Surface synoptic meteorology reports
TAF	Terminal Aerodrome Forecasts
UAR	Upper Air Reports

The remaining parameters (Sections A.2 through A.7) are used only when grib is specified as the product-id.

## A.8 layer

**Table A-3. layer parameter codes**

layer	Description	Example
adiabatic-cond	Adiabatic condensation level (parcel lifted from surface)	(layer adiabatic-cond)
atm-top	Level of the top of the atmosphere	(layer atm-top)
cloud-base	Cloud base level	(layer cloud-base)
cloud-top	Cloud top level	(layer cloud-top)
conv-cld-base	Level of bases of convective clouds	(layer conv-cld-base)
conv-cld-top	Level of tops of convective clouds	(layer conv-cld-top)
entire-atm	Entire atmosphere	(layer entire-atm)

**Table A-3. layer parameter codes**

<b>layer</b>	<b>Description</b>	<b>Example</b>
entire-ocean	Entire ocean	(layer entire-ocean)
height	Height above ground (meters)	(layer height 1500)
height-between	Layer between two heights above ground in hundreds meters (followed by top and bottom level values)	(layer height-between 50 30) for layer between 5000 and 3000 meters above ground
height-between-ft	Layer between two heights above ground, in feet (followed by top and bottom level values)	(layer height-between-ft 15000 10000)
height-ft	Height above ground (feet)	(layer height-ft 50)
high-cld-base	Level of high cloud bases	(layer high-cld-base)
high-cld-top	Level of high cloud tops	(layer high-cld-top)
hybrid	Hybrid level (followed by level number)	(layer hybrid 1)
hybrid-between	Layer between two hybrid levels (followed by top and bottom level numbers)	(layer hybrid 2 1)
isobar	Level of an isobaric surface (followed by the isobar value of the surface in hectoPascals (hPa) (1000, 975, 950, 925, 900, 850, 800, 750, 700, 650, 600, 550, 500, 450, 400, 350, 300, 250, 200, 150, 100, 70, 50, 30, 20, 10))	(layer isobar 500)
isobar-between	Layer between two isobaric surfaces (followed by top and bottom isobar values in kPa, separated by a space)	(layer isobar-between 50 100) for layer between 500 and 1000 hPa
isobar-between-mp	Layer between two isobaric surfaces, mixed precision (followed by pressure of top in kPa and 1100 minus pressure of bottom in hPa)	(layer isobar-between-mp 50 100) for layer between 500 and 1000 hPa
isobar-between-xp	Layer between two isobaric surfaces, extra precision (followed by top and bottom isobar values expressed as 1100 hPa-isobar level, separated by a space)	(layer isobar-between 600 100) for layer between 500 and 1000 hPa

**Table A-3. layer parameter codes**

layer	Description	Example
isotherm-0	Level of the zero-degree (Celsius) isotherm (or freezing level)	(layer isotherm-0)
land-depth	Depth below land surface in centimeters	(layer land-depth 5.0)
land-depth-between	Layer between two depths in ground (followed by the depth of the top of the layer and the depth of the bottom of the layer centimeters)	(layer land-depth-between 0 30) for layer from ground surface to 30 cm depth
land-height-cm	Height level above ground (high precision) (followed by height in centimeters)	(layer land-height-cm 50)
land-isobar	Pressure above ground level in hPa	(layer land-isobar 500)
land-isobar-between	Layer between two isobars above levels (followed by top and bottom isobaric levels in hPa)	(layer land-isobar-between 500 1000)
low-cld-base	Level of low cloud bases	(layer low-cld-base)
low-cld-top	Level of low cloud tops	(layer low-cld-top)
max-wind	Level of maximum wind	(layer max-wind)
mid-cld-base	Level of middle cloud bases	(layer mid-cld-base)
mid-cld-top	Level of middle cloud tops	(layer mid-cld-top)
msl	Mean sea level	(layer msl)
msl-height	Height above mean sea level (in meters)	(layer msl-height 50)
msl-height-between	Layer between two heights above mean sea level in hundreds of meters (followed by top and bottom height values)	(layer msl-height-between 10 5) for layer between 1000 and 500 meters above ground
msl-height-ft	Height above mean sea level (in feet)	(layer msl-height-ft 5000)
sea-bottom	Bottom of the ocean	(layer sea-bottom)
sea-depth	Depth below the sea surface (meters)	(layer sea-depth 50)
sigma	Sigma level in 1/10000	(layer sigma 9950) for sigma level .995

**Table A-3. layer parameter codes**

<b>layer</b>	<b>Description</b>	<b>Example</b>
sigma-between	Layer between two sigma surfaces (followed by top and bottom sigma values expressed in 1/100, separated by a space)	(layer sigma-between 99.5 100.0) for layer between .995 and 1.0
sigma-between-xp	Layer between two sigma levels (followed by top and bottom sigma values expressed as 1.1-sigma)	(layer sigma-between-xp .105 .100) for layer between .995 and 1.0
surface	Earth's surface	(layer surface)
theta	Isentropic (theta) level (followed by potential temperature in degrees K)	(layer theta 300)
theta-between	Layer between two isentropic surfaces (followed by top and bottom values expressed as 475-theta in degrees K)	(layer theta-between 150 200)
tropopause	Level of tropopause (top of troposphere)	(layer tropopause)

## A.9 model

**Table A-4. model parameter codes**

<b>Model Name</b>	<b>Description</b>	<b>Process ID</b>	<b>Grid ID</b>
3D_cent_am_nest2_appl_22km			
3D_e_pac_nest2_appl_22km			
3D_europe_nest2_appl2_22km			
3D_europe_nest3_appl5_11km			
3D_gin_sea_306x101_22km			
3D_global_360x181_111km			
3D_ind_ocn_106x111_111km			
3D_korea_nest2_appl2_22km			
3D_n_am_81x61_111km			
3D_n_atl_ocn_171x96_111km			
3D_n_atl_ocn_171x96_111km			
3D_n_pac_ocn_191x86_111km			
3D_s_atl_ocn_106x91_111km			

**Table A-4. model parameter codes**

Model Name	Description	Process ID	Grid ID
3D_s_pac_ocn_196x91_111km			
3D_southwest_asia_11km			
3D_southwest_asia_22km			
3D_w_atl_nest2_appl_22km			
3D_w_atl_ocn_211x176_22km			
3D_w_pac_ocn_201x201_22km			
COAMPS	Coupled Ocean-Atmosphere Mesoscale Prediction System	22	183
COAMPS_Eur	Coupled Ocean-Atmosphere Mesoscale Prediction System - Europe		
COAMPS_Korea	Coupled Ocean-Atmosphere Mesoscale Prediction System - Korea		
DAF			
ECMWF-1deg		48	240
ECMWF-48-240	European Centre for Mid-Range Weather Forecasting		
EOTS		40	223
Flight_Level	Flight level temperatures and winds	10	240
FNMOC_23_240			
FNMOC_47_240			
FNMOC_50_240			
GOXM		83	223
GSOWM		17	240
NCEP_MRF_62Wave_2_2.5Deg			
NMC		78	-9999
NOGAPS	Navy Operational Global Atmospheric Prediction System	58	223
NOGAPS-1deg	NOGAPS 1 degree grid	58	240
NORAPS-Asia	Navy Operational Regional Atmospheric Prediction System - Asia	70	242

**Table A-4. model parameter codes**

<b>Model Name</b>	<b>Description</b>	<b>Process ID</b>	<b>Grid ID</b>
NORAPS-Bosnia	Navy Operational Regional Atmospheric Prediction System - Bosnia	76	249
NORAPS-CONUS	Navy Operational Regional Atmospheric Prediction System - Continental US	71	245
NORAPS-Europe	Navy Operational Regional Atmospheric Prediction System - Europe	72	244
NORAPS-Indian-Ocean	Navy Operational Regional Atmospheric Prediction System - Indian Ocean	74	243
NORAPS-SoCal	Navy Operational Regional Atmospheric Prediction System - Southern California	75	237
OTIS		43	240
STRATOI		61	240
TOPS		28	240
TROPO		82	223
TYAN	Typhoon Analog	68	223
UANVA		65	223
WAM	Wave Action Model	17	240
WAM_Korea	Wave Action Model - Korea		

## A.10 product-GRIB-code

Gridded data products may be specified either by keyword or by product-GRIB-code. The product-GRIB-code parameter allows a shorthand specification of the product. The following GRIB codes and notes are taken from *WMO World Weather Watch Technical Report 17, Guide to WMO Binary Code Forms*, Part 2, Table 2.

**Table A-5. product-GRIB-code and product keyword values**

<b>Value</b>	<b>Keyword</b>	<b>Description</b>	<b>Units</b>
000		Reserved	
001	pressure	Pressure	Pa
002	pressure-msl	Pressure reduced to MSL	Pa
003	pressure-tendency	Pressure tendency	Pa/s

**Table A-5. product-GRIB-code and product keyword values**

<b>Value</b>	<b>Keyword</b>	<b>Description</b>	<b>Units</b>
004			
005			
006	geopotential	Geopotential	$\text{m}^2/\text{s}^2$
007	geopotential-height	Geopotential height	Gpm
008	geometric-height	Geometric height	M
009	standard-deviation-height	Standard deviation of height	M
010	total-ozone		
011	temperature	Temperature	K
012	virtual-temperature	Virtual temperature	K
013	potential-temperature	Potential temperature	K
014	pseudo-adiabatic-potential-temperature	Pseudo-adiabatic potential temperature	K
015	maximum-temperature	Maximum temperature	K
016	minimum-temperature	Minimum temperature	K
017	dew-point-temperature	Dew point temperature	K
018	dew-point-depression	Dew point depression (or deficit)	K
019	lapse-rate	Lapse rate	K/m
020	visibility	Visibility	M
021	radar-spectra	Radar Spectra (1)	-
022	radar-spectra	Radar Spectra (2)	-
023	radar-spectra	Radar Spectra (3)	-
024	parcel-lifted-index		
025	temperature-anomaly	Temperature anomaly	K
026	pressure-anomaly	Pressure anomaly	Pa
027	geopotential-height-anomaly	Geopotential height anomaly	Gpm
028	wave-spectra	Wave Spectra (1)	-
029	wave-spectra	Wave Spectra (2)	-
030	wave-spectra	Wave Spectra (3)	-
031	wind-direction	Wind direction	Deg. true
032	wind-speed	Wind speed	m/s
033	u-component-wind	u-component of wind	m/s
034	v-component-wind	v-component of wind	m/s
035	stream-function	Stream function	$\text{m}^2/\text{s}$

**Table A-5. product-GRIB-code and product keyword values**

<b>Value</b>	<b>Keyword</b>	<b>Description</b>	<b>Units</b>
036	velocity-potential	Velocity potential	m <sup>2</sup> /s
037	montgomery-stream-function	Montgomery stream function	m <sup>2</sup> /s <sup>2</sup>
038	sigma-coord-vertical-velocity	Sigma coord. vertical velocity	s /s
039	pressure-vertical-velocity	Pressure Vertical velocity	Pa/s
040	geometric-vertical-velocity	Geometric Vertical velocity	m/s
041	absolute-vorticity	Absolute vorticity	/s
042	absolute-divergence	Absolute divergence	/s
043	relative-vorticity	Relative vorticity	/s
044	relative-divergence	Relative divergence	/s
045	vertical-u-component-shear	Vertical u-component shear	/s
046	vertical-v-component-shear	Vertical v-component shear	/s
047	direction-of-current	Direction of current	Deg. true
048	speed-of-current	Speed of current	m/s
049	u-component-current	u-component of current	m/s
050	v-component-current	v-component of current	m/s
051	specific-humidity	Specific humidity	kg/kg
052	relative-humidity	Relative humidity	%
053	humidity-mixing-ratio	Humidity mixing ratio	kg/kg
054	precipitable-water	Precipitable water	kg/m <sup>2</sup>
055	vapor-pressure	Vapor pressure	Pa
056	saturation-deficit	Saturation deficit	Pa
057	evaporation	Evaporation	kg/m <sup>2</sup>
058	cloud-ice	Cloud Ice	kg/m <sup>2</sup>
059	precipitation-rate	Precipitation rate	kg/m <sup>2</sup> /s
060	thunderstorm-probability	Thunderstorm probability	%
061	total-precipitation	Total precipitation	kg/m <sup>2</sup>
062	large-scale-precipitation	Large scale precipitation	kg/m <sup>2</sup>
063	convective-precipitation	Convective precipitation	kg/m <sup>2</sup>
064	snowfall-rate-water-equivalent	Snowfall rate water equivalent	kg/m <sup>2</sup> s
065	water-equiv-accum-snow-depth	Water equiv. of accum. snow depth	kg/m <sup>2</sup>
066	snow-depth	Snow depth	M
067	mixed-layer-depth	Mixed layer depth	M

**Table A-5. product-GRIB-code and product keyword values**

<b>Value</b>	<b>Keyword</b>	<b>Description</b>	<b>Units</b>
068	transient-thermocline-depth	Transient thermocline depth	M
069	main-thermocline-depth	Main thermocline depth	M
070	main-thermocline-anomaly	Main thermocline anomaly	M
071	total-cloud-cover	Total cloud cover	%
072	convective-cloud-cover	Convective cloud cover	%
073	low-cloud-cover	Low cloud cover	%
074	medium-cloud-cover	Medium cloud cover	%
075	high-cloud-cover	High cloud cover	%
076	cloud-water	Cloud water	kg/m <sup>2</sup>
077	best-lifted-index		
078	convective-snow	Convective snow	kg/m <sup>2</sup>
079	large-scale-snow	Large scale snow	kg/m <sup>2</sup>
080	water-temperature	Water Temperature	K
081	land-sea-mask	Land-sea mask (1=land;0=sea) (see note)	Fraction
082	deviation-of-sea-level-from-mean	Deviation of sea level from mean	M
083	surface-roughness	Surface roughness	M
084	albedo	Albedo	%
085	soil-temperature	Soil temperature	K
086	soil-moisture-content	Soil moisture content	kg/m <sup>2</sup>
087	vegetation	Vegetation	%
088	salinity	Salinity	kg/kg
089	density	Density	kg/m <sup>3</sup>
090	water-runoff	Water run off	kg/m <sup>2</sup>
091	ice-concentration	Ice concentration (ice=1;no ice=0)(see note)	Fraction
092	ice-thickness	Ice thickness	M
093	direction-of-ice-drift	Direction of ice drift	deg. true
094	speed-of-ice-drift	Speed of ice drift	m/s
095	u-component-of-ice-drift	u-component of ice drift	m/s
096	v-component-of-ice-drift	v-component of ice drift	m/s
097	ice-growth-rate	Ice growth rate	m/s
098	ice-divergence	Ice divergence	/s

**Table A-5. product-GRIB-code and product keyword values**

<b>Value</b>	<b>Keyword</b>	<b>Description</b>	<b>Units</b>
099	snow-melt	Snow melt	kg/m <sup>2</sup>
100	significant-height-waves-swell	Significant height of combined wind waves and swell	m
101	direction-wind-waves	Direction of wind waves	deg. true
102	significant-height-wind-waves	Significant height of wind waves	m
103	mean-period-wind-waves	Mean period of wind waves	s
104	direction-swell-waves	Direction of swell waves	deg. true
105	significant-height-swell-waves	Significant height of swell waves	m
106	mean-period-swell-waves	Mean period of swell waves	s
107	primary-wave-direction	Primary wave direction	deg. true
108	primary-wave-mean-period	Primary wave mean period	s
109	secondary-wave-direction	Secondary wave direction	deg. true
110	secondary-wave-mean-period	Secondary wave mean period	s
111	net-short-wave-radiation-surface	Net short-wave radiation (surface)	W/m <sup>2</sup>
112	net-long-wave-radiation-surface	Net long wave radiation (surface)	W/m <sup>2</sup>
113	net-short-wave-radiation-top-of-atmos	Net short-wave radiation (top of atmos.)	W/m <sup>2</sup>
114	net-long-wave-radiation-top-of-atmos	Net long wave radiation (top of atmos.)	W/m <sup>2</sup>
115	long-wave-radiation	Long wave radiation	W/m <sup>2</sup>
116	short-wave-radiation	Short wave radiation	W/m <sup>2</sup>
117	global-radiation	Global radiation	W/m <sup>2</sup>
118	brightness-temperature		
119	long-wave-radiation		
120	short-wave-radiation		
121	latent-heat-net-flux	Latent heat net flux	W/m <sup>2</sup>
122	sensible-heat-net-flux	Sensible heat net flux	W/m <sup>2</sup>
123	boundary-layer-dissipation	Boundary layer dissipation	W/m <sup>2</sup>
124	momentum-flux-u-component	Momentum flux, u component	N/m <sup>2</sup>
125	momentum-flux-v-component	Momentum flux, v component	N/m <sup>2</sup>
126	wind-mixing-energy	Wind mixing energy	J
127	image-data	Image data	
128-141		Reserved for use by originating	

**Table A-5. product-GRIB-code and product keyword values**

<b>Value</b>	<b>Keyword</b>	<b>Description</b>	<b>Units</b>
		center	
142	ditch-headings	Ditch Headings	degrees
143-154		Reserved for use by originating center)	
155	white-cap-probability	Whitecap probability	percent
156-174		Reserved for use by originating center	
175	evaporative-duct-height	Evaporative duct height	meters
176-204		Reserved for use by originating center	
205	sonic-layer-depth	Sonic layer depth	meters
206-254		Reserved for use by originating center	
255		Missing	

**Notes:**

1. By convention, downward net fluxes of radiation or other quantities are assigned negative values; upward net fluxes of radiation or other quantities are assigned positive values.
2. Unidirectional flux values, where the direction of flow is indicated in the name of the parameter shall all have positive values irrespective of the direction of flow. Net (vertical) fluxes shall be calculated by subtracting the downward flux values from the upward flux values
3. The u and v components of vector quantities are defined with reference to GDS Octet 17 and Table 7. However, if the GDS is **not** included in a message, then any wind components are assumed to be resolved relative to the grid specified in the PDS with u and v defined as positive in the direction of increasing x and y (or i and j) coordinates respectively.
4. Provision is made for three types of spectra:
  - 1) Direction and Frequency
  - 2) Direction and radial number
  - 3) Radial number and radial number

5. Parameters 81 and 91 show the units as "fraction", thus allowing for a range of coverage. It is up to the user to employ the D (power of ten) scaling to assure that the necessary precision is retained in the numeric values.

## A.11 tau

The `tau` field contains a list of forecast hours desired, separated by spaces. For example, (`tau 0 12 24 48 72`) would be used to request the analysis product (`tau 0`) and the 12, 24, 48, and 72 hour forecasts.

## A.12 resolution

The `resolution` field contains the desired resolution of gridded data in degrees. For example (`resolution 2.5`) specifies 2.5 degree grid spacing.

This page intentionally left blank.

## Appendix B                    THE COMPLETE OMF DTD

```
<!-- <!DOCTYPE OMF SYSTEM "OMF.dtd" [ -->
<!-- Weather Observation Definition Format DTD -->
<!-- This is the OMF XML DTD. It can be referred to using the
formal public identifier
     -//METNET//OMF 1.0//EN
For description, see OMF.html

$Id: OMF.dtd,v 3.8 1999/10/25 18:18:31 oleg Exp oleg $
-->

<!-- Weather Observation Definition Format -->

<!-- Basic attributes -->

<!ENTITY % TStamp-type "NMTOKEN">
<!ENTITY % TRange-type "CDATA">
<!ENTITY % TStamp "TStamp %TStamp-type; #REQUIRED">
<!ENTITY % TRange "TRange %TRange-type; #REQUIRED">
<!ENTITY % LatLon "LatLon CDATA #REQUIRED">
<!ENTITY % LatLons "LatLons CDATA #REQUIRED">
<!ENTITY % BBox-REQD "BBox CDATA #REQUIRED">
<!ENTITY % BBox-OPT "BBox CDATA #IMPLIED">
<!ENTITY % BId "BId NMTOKEN #REQUIRED">
<!ENTITY % SName "SName CDATA #REQUIRED">
<!ENTITY % Elev "Elev NMTOKEN #IMPLIED">

<!-- Basic elements -->

<!ELEMENT VALID (#PCDATA)>
<!ATTLIST VALID %TRange;>

<!-- A collection of weather observation reports -->

<!ELEMENT Reports ( METAR | SPECI | PIREP | UAR | BTSC | SYN )*>
<!ATTLIST Reports %TStamp;>

<!-- Common report attributes -->
<!ENTITY % ReportAttrs
  "%TStamp; %LatLon; %BId; %SName; %Elev;
  Vis NMTOKEN #IMPLIED
  Ceiling NMTOKEN #IMPLIED
  ">

<!-- METAR and SPECI reports -->

<!ELEMENT METAR (#PCDATA)>
<!ATTLIST METAR %ReportAttrs;>

<!ELEMENT SPECI (#PCDATA)>
<!ATTLIST SPECI %ReportAttrs;>

<!-- A collection of weather hazard advisories -->

<!ELEMENT Advisories ( SIGMET | AIRMET | WW )* >
```

```
<!ATTLIST Advisories %TStamp;>

<!-- A SIGMET advisory -->

<!ELEMENT SIGMET (VALID, AFFECTING?, EXTENT, BODY) >
<!ATTLIST SIGMET
    class (CONVECTIVE| HOTEL| INDIA| UNIFORM| VICTOR| WHISKEY) #REQUIRED
    id NMTOKEN #REQUIRED
    %TStamp;
    %BBox-OPT;
>

<!ELEMENT AFFECTING (#PCDATA)>

<!ELEMENT EXTENT (#PCDATA)>
<!ATTLIST EXTENT
    Shape (AREA| LINE| POINT) #REQUIRED
    %LatLons;
>

<!ELEMENT BODY (#PCDATA)>

<!-- A AIRMET advisory -->

<!ELEMENT AIRMET (HEADER?, VALID, AIPART+)>
<!ATTLIST AIRMET
    class (SIERRA|TANGO|ZULU)#REQUIRED
    %Tstamp NMTOKEN #REQUIRED
    %Sname NMTOKEN #REQUIRED
    %Bid NMTOKEN #REQUIRED
    %WMOHeader; NMTOKEN #REQUIRED
>

<!ELEMENT HEADER (#PCDATA)
<!ATTLIST HEADER
    %Update NMTOKEN #IMPLIED
>

<!ELEMENT VALID (#PCDATA)
<!ATTLIST VALID
    %TRange CDATA #REQUIRED
>

<!ELEMENT AIPART (TYPE?, AFFECTING?, EXTENT? BODY) >
<!ELEMENT TYPE (#PCDATA) >
<!ELEMENT AFFECTING (#PCDATA) >
<!ELEMENT EXTENT (#PCDATA)>
<!ATTLIST EXTENT
    %LatLons CDATA #REQUIRED
<!ELEMENT BODY (#PCDATA)>

>

<!-- A collection of weather forecasts -->
```

```
<!ELEMENT Forecasts ( TAF )* >
<!ATTLIST Forecasts %TStamp;>

<!-- A Terminal Aerodrome Forecast -->

<!ELEMENT TAF ( VALID, PERIOD+ ) >
<!ATTLIST TAF
  %TStamp; %LatLon; %BId; %SName;
>

<!ELEMENT PERIOD ( PREVAILING, VAR* )>
<!ATTLIST PERIOD
  %TRange;
  Title NMTOKEN #IMPLIED
>

<!ELEMENT PREVAILING (#PCDATA)>

<!ELEMENT VAR (#PCDATA)>
<!ATTLIST VAR
  %TRange;
  Title CDATA #REQUIRED
>

<!-- Rawinsonde and Pibal Observation reports -->

<!ELEMENT UAR ( UAPART+, UAID*, UACODE*, UALEVELS ) >
<!ATTLIST UAR
  %TStamp; %LatLon; %BId; %SName; %Elev;
>

<!ELEMENT UAPART (#PCDATA)>
<!ATTLIST UAPART
  id NMTOKEN #REQUIRED
>
<!ENTITY % UARef "Ref NMTOKEN #REQUIRED">

<!ELEMENT UAID (#PCDATA)>
<!ATTLIST UAID %UARef; >
<!ELEMENT UACODE (#PCDATA)>
<!ATTLIST UACODE %UARef; >

<!ELEMENT UALEVELS (UALEVEL)*>
<!ELEMENT UALEVEL (#PCDATA)>
<!ATTLIST UALEVEL
  %UARef;
  P NMTOKEN #REQUIRED
  H NMTOKEN #IMPLIED
  T NMTOKEN #IMPLIED
  DP NMTOKEN #IMPLIED
  Wind CDATA #IMPLIED
>

<!-- Bathythermal, Salinity and Ocean Currents Observations -->
```

```
<!ELEMENT BTSC ( BTID, BTCODE?, BTLEVELS ) >
<!ATTLIST BTSC
  %TStamp; %LatLon; %BId; %SName;
  Title (JJYY | KKXX | NNXX) #REQUIRED
  Depth NMOKEN #IMPLIED
>

<!ELEMENT BTID (#PCDATA)>
<!ATTLIST BTID
  DZ (7|8) #IMPLIED
  Rec NMOKEN #IMPLIED
  WS (0|1|2|3) #IMPLIED
  Curr-s (2|3|4) #IMPLIED
  Curr-d NMOKEN #IMPLIED
  AV-T (0|1|2|3) #IMPLIED
  AV-Sal (0|1|2|3) #IMPLIED
  AV-Curr (0|1|2|3) #IMPLIED
  Sal (1|2|3) #IMPLIED
>

<!ELEMENT BTCODE (#PCDATA)>
<!ELEMENT BTLEVELS (BTAIR?, (BTLEVEL)*)>

<!ELEMENT BTAIR (#PCDATA)>
<!ATTLIST BTAIR
  T NMOKEN #IMPLIED
  Wind CDATA #IMPLIED
>

<!ELEMENT BTLEVEL (#PCDATA)>
<!ATTLIST BTLEVEL
  D NMOKEN #REQUIRED
  T NMOKEN #IMPLIED
  S NMOKEN #IMPLIED
  Curr CDATA #IMPLIED
>

<!-- Surface Synoptic Reports from land and sea stations -->

<!ELEMENT SYN ( SYID, SYCODE?, SYG?, SYSEA? ) >
<!ATTLIST SYN
  %TStamp; %LatLon; %BId; %SName; %Elev;
  Title (AAXX | BBXX | ZZYY) #REQUIRED
  SType (AUTO | MANN) "MANN"
>

<!ELEMENT SYID (#PCDATA)>
<!ATTLIST SYID
  WS (0|1|3|4) #IMPLIED
>

<!ELEMENT SYCODE (#PCDATA)>

<!ELEMENT SYG (#PCDATA)>
<!ATTLIST SYG
  T NMOKEN #IMPLIED
  TD NMOKEN #IMPLIED
```

```
Hum NMTOKEN #IMPLIED
Tmm CDATA #IMPLIED
P NMTOKEN #IMPLIED
PO NMTOKEN #IMPLIED
Pd NMTOKENS #IMPLIED
Vis NMTOKEN #IMPLIED
Ceiling NMTOKEN #IMPLIED
Wind CDATA #IMPLIED
WX CDATA #IMPLIED
Prec CDATA #IMPLIED
Clouds CDATA #IMPLIED
>

<!ELEMENT SYSEA (#PCDATA)>
<!ATTLIST SYSEA
  T NMTOKEN #IMPLIED
  Wave CDATA #IMPLIED
  SDir CDATA #IMPLIED
>

<!-- A PIREP Report -->

<!ELEMENT PIREP(PIID, PIC, PIRMK)>
<!ATTLIST PIREP
  %Rtype (ROUTINE|URGENT) #REQUIRED
  %Tstamp NMTOKEN #REQUIRED
  %LatLon CDATA #REQUIRED
  %Bid NMTOKEN #REQUIRED
  %Sname CDATA #REQUIRED
  %WMOHeader NMTOKEN #REQUIRED
  %Flevel NMTOKEN #IMPLIED
>

<!ELEMENT PIID (#PCDATA) >
<!ATTLIST PIID
  ATType CDATA #IMPLIED
>
<!Element PIC (#PCDATA)>
<!ATTLIST PIC
  T NMTOKEN #IMPLIED
  Wind CDATA #IMPLIED
>

<!ELEMENT PIRMKS ( PIWX?, PISKY?, PITB?, PIICE?, PIRMK? )>
<!ELEMENT PIWX (#PCDATA)>
<!ELEMENT PISKY (#PCDATA)>
<!ELEMENT PIICE (#PCDATA)>
<!ELEMENT PIRMKS (#PCDATA)>

<!-- Plain-text WMO Meteorological messages -->

<!ELEMENT Messages ( MSG )* >
<!ATTLIST Messages %TStamp;>

<!ELEMENT MSG ANY >
<!ATTLIST MSG
  id NMTOKEN #REQUIRED
```

## **fnmoc\_metcast\_pg**

---

```
Type      NMTOKEN #IMPLIED
%TStamp;
%SName;
%BBox-OPT;
BBB      CDATA   #IMPLIED
Descr    CDATA   #IMPLIED
>

<!-- ]> -->
```

## Appendix C

## SAMPLE JAVA PROGRAM FOR DATA RETRIEVAL

```
import java.io.*;
import java.util.*;
import omnicast.retrieverService.*;
import omnicast.util.Environment;
import omnicast.shared.*;
import java.rmi.registry.*;
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.NotBoundException;
import java.rmi.RMISecurityManager;
import java.net.MalformedURLException;

/* Sample program that communicates with the Retriever Service */
public class RetrieverSample {

    public static void main(String[] args){
        RetrieverServiceInterface serviceObj; // Handle to Remote Retriever
        String localHost; // Local IP Host
        int rmiPort; // Socket Port for Connection
        String rmiPortStr; // String representation
        String connectURL; // URL for remote Retriever

        // Use omnicast utility library to get RMI port setting from
        // the .rsettings file located in current directory.
        rmiPort = omnicast.util.Environment.getRmiPort();
        rmiPortStr = Integer.toString(rmiPort); // Convert to String
        try {
            // Get the IP host for this computer
            localHost=java.net.InetAddress.getLocalHost().toString();
            // Remove extra garbage in name that Windows OS's might return
            localHost=omnicast.winutil.Util.truncateHostName(localHost);
            // Compose the URL for connecting to the Retriever Service
            // Result will look like "//myHost:1099/RetrieverService"
            connectURL="//"+localHost+":"+rmiPortStr+"/RetrieverService";
            // Make the connection to the Retriever Service
            serviceObj=(RetrieverServiceInterface)Naming.lookup(connectURL);
            // Got here so we must have success
            System.out.println("Connection to the Retriever Service "+"was
            successful");

            // Create a Retriever Session that has an AREA NAME of myArea, a
            // DATA DOMAIN of gridded, and a SERVER NAME of FNMOC.
            RetrieverSessionInterface remoteSession;
            remoteSession=serviceObj.createRetrieverSession("myArea","gridded",
                "FNMOC");

            // Configure the Retriever Session's Properties
            remoteSession.setNetDownAllowanceEnabled(false);
            remoteSession.setReqIfModifiedEnabled(true);
            remoteSession.setReqIfModifiedSince(720);
            remoteSession.setRequestMimeType("text/x-mbl");
        }
    }
}
```

```
remoteSession.setRetryOnErrorCountEnabled(true);
remoteSession.setRetryOnErrorCount(5);
remoteSession.setReqUpdatesEnabled(true);
remoteSession.setReqUpdatesInterval(300);      // Check every 5 minutes
remoteSession.setServerUrl("http://152.80.49.210/cgi-
bin/mcsrvr/rest/server");
remoteSession.setMetcastVersion("RetrieverSample");

// Configure to use Authentication
String authCredential="Basic dIWkIkkHSiOOikOw";
// String authCredential = "Basic " + computeCookie("username",
"password");
remoteSession.setAuthorizationEnabled(true);
remoteSession.setAuthCredential(authCredential);

// Set the Server Proxy Settings
// remoteSession.setProxyEnabled(true); // Set to false if you don't
// have a proxy
// remoteSession.setProxyName("192.16.167.45"); // Replace with your
// proxy's IP address here
// remoteSession.setProxyPort(80);
// remoteSession.setProxyAuth("Basic DxkdDiASHIO"); // Proxy
// authorization, if any

// Define the Request Message String
String requestMessageString =
  "(MYAREA\n"+
  " (bounding-box 57.5 -112.5    7.5   -30.0)\n"+
  " (products\n"+
  "  (grib\n"+
  "   (bounding-box 57.0 -112.0    8.0   -32.0)\n"+
  "   (product-GRIB-code 11) (layer height-ft 1000) (tau 12 36) "+
  "  (source 58) (process-id 10) (resolution 1.000))\n"+
  " )\n";
remoteSession.setRequestString(requestMessageString);

// Configure a Mailcap for the retrieval Session
MailcapInterface remoteMailcap;
MailcapEntryInterface remoteMailcapEntry;
MimeOperationInterface remoteMimeOperation;
String operationString = "copy %s swh.grib";
StringTokenizer st;

// Get the Mailcap
remoteMailcap = (MailcapInterface)remoteSession.getMailcap();
// Create a Mailcap Entry in our Mailcap
remoteMailcapEntry =
(MailcapEntryInterface)remoteMailcap.createMailcapEntry();
// Set the MIME type for this Mailcap entry
remoteMailcapEntry.setMimeType("application/x-grib");
// An operation will be performed after data of this MIME type is
downloaded
remoteMailcapEntry.setOperationEnabled(true);
// No operation will occur before data of this MIME type is downloaded
remoteMailcapEntry.setPreOperationEnabled(false);
// No operation will occur after all data of this MIME type is
// downloaded
```

## **fnmoc\_metcast\_pg**

---

```
remoteMailcapEntry.setPostOperationEnabled(false);
// Create a MIME Operation object
remoteMimeOperation =
//(MimeOperationInterface)remoteMailcapEntry.getOperation();
// Enable this MIME operation
remoteMimeOperation.setOperationEnabled(true);
// Set the arguments of this MIME operation by parsing the
//operationString
st = new StringTokenizer(operationString);
while (st.hasMoreTokens()){
    remoteMimeOperation.addArgument(st.nextToken());
}

// Starting the Retrieval Session
System.out.println("Starting retrieval session");
remoteSession.startSession();

}catch (Exception e){
    // Exception occurred. Most likely the Retriever Service is not
    // running.
    System.out.println("Could not connect to Retriever "+
                       "Service\n"+e.getMessage());
}
}
```

This page intentionally left blank.

## Appendix D            CODE TO COMPUTE AN AUTHORIZATION CREDENTIAL

The following code should be appended to the sample program in Appendix C above, if computation of an authorization credential is desired.

```
protected static final String baseChars =
    "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/" ;

/*      Compute HTTP 1.1 Basic Authorization credentials */

/* Encode str in BASE64 [RFC2045] */
/* The result is a single continuous string of BASE64 characters */
/* (not broken in any way by newlines) */
public static String base64Encode(final String inStr) {
    int nLength = inStr.length();
    int nOutLength = nLength * 4 / 3 + 1;

    if ( ( nLength * 4 % 3 ) != 0 )
        nOutLength += 3 - ( nLength * 4 ) % 3;
    StringBuffer outStr = new StringBuffer(nOutLength);

    for (int i = 0; i < nLength; ) {
        int nBuffer = (int) inStr.charAt(i++);      /* 8 bits in nBuffer */
        outStr.append(baseChars.charAt((nBuffer >> 2) & 63));
        if ( i >= nLength ) {
            outStr.append(baseChars.charAt((nBuffer << 4) & 63)); /* pad by 4
zero bits */
            outStr.append('=');
            outStr.append('=');
            break;
        }

        nBuffer = (nBuffer << 8) | (int) inStr.charAt(i++); /* 10 bits to
encode */
        outStr.append(baseChars.charAt((nBuffer >> 4) & 63));
        if( i >= nLength ) {
            outStr.append(baseChars.charAt((nBuffer << 2) & 63)); /* pad by 2
zero bits */
            outStr.append('=');
            break;
        }

        nBuffer = (nBuffer << 8) | (int) inStr.charAt(i++); /* 12 bits to
encode */
        outStr.append(baseChars.charAt((nBuffer >> 6) & 63));
        outStr.append(baseChars.charAt(nBuffer & 63));
    }

    return outStr.toString();
}

/* decode string from BASE64 */
```

```
public static String base64Decode(final String inStr) {
    StringBuffer outStr = new StringBuffer();
    int nBuffer;

    int i = 0;
    while (i < inStr.length()) {
        int pstr = baseChars.indexOf(inStr.charAt(i++));
        if (pstr < 0)
            return "";

        nBuffer = pstr; /* 6 bits in nBuffer */
        pstr = baseChars.indexOf(inStr.charAt(i++));
        if (pstr < 0)
            return "";

        nBuffer = nBuffer << 6 | pstr; /* 12 bits in nBuffer */
        outStr.append((char) ((nBuffer >> 4) & 0xff)); /* now 4 bits in
nBuffer */

        if (inStr.charAt(i) == '=' && inStr.charAt(i+1) == '=') {
            break; /* we're done */
        }

        pstr = baseChars.indexOf(inStr.charAt(i++));
        if (pstr < 0)
            return "";

        nBuffer = (nBuffer & 15) << 6 | pstr; /* 10 bits in nBuffer */
        outStr.append((char) ((nBuffer >> 2) & 0xff));

        if (inStr.charAt(i) == '=') {
            break; /* we're done */
        }

        pstr = baseChars.indexOf(inStr.charAt(i++));
        if (pstr < 0)
            return "";

        nBuffer = (nBuffer & 3) << 6 | pstr; /* 8 bits in nBuffer */
        outStr.append((char) (nBuffer & 0xff));
    }

    return outStr.toString();
}

/* Compute an authorization cookie as outlined in the algorithm above */
public static String computeCookie(final String userId, final String
passwd) {
    return base64Encode(userId + ":" + passwd);
}
```